RI/RD96-182

Comparative Study of Advanced Turbulence Models for Turbomachinery

Ali H. Hadid and Munir M. Sindir

Rocketdyne Division
Boeing North American Inc.
CFD Technology Center
Canoga Park, CA 91309

Contract NAS8-38860
Final Report

Approved by

Glenn Havskjold

Glenn Havskjold
Program Manager

October 1996

Comparative Study of Advanced Turbulence Models for Turbomachinery

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

This work was supported jointly by NASA Marshall Space Flight Center under contract NAS8-38860 and by Rocketdyne Division of Rockwell International under discretionary funds. The authors gratefully acknowledge the efforts of NASA contract monitors Dr. Paul McConnaughey and Mr. Robert Garcia and Rocketdyn program manager Dr. Glenn Havskjold. The authors would also like to acknowledge the efforts of Professor C.-P. Chen and his colleagues Dr. Hong Wei and Ms. X. Li at the University of Alabama at Huntsville for their help in testing the modules. The assistance of Mr. Steve Barson in managing the project and the efforts of Ms. Michelle DeCroix and Mr. Armen Darian of Rocketdyne in testing some of the modules are greatly appreciated.

SUMMARY

A computational study has been undertaken to study the performance of advanced phenomenological turbulence models coded in a modular form to describe incompressible turbulent flow behavior in two dimensional/axisymmetric and three dimensional complex geometry. The models include a variety of two equation models (single and multi-scale k-ε models with different near wall treatments) and second moment algebraic and full Reynolds stress closure models. These models were systematically assessed to evaluate their performance in complex flows with rotation, curvature and separation. The models are coded as self contained modules that can be interfaced with a number of flow solvers. These modules are stand alone satellite programs that come with their own formulation, finite-volume discretization scheme, solver and boundary condition implementation. They will take as input (from any generic Navier-Stokes solver) the velocity field, grid (structured H-type grid) and computational domain specification (boundary conditions), and will deliver, depending on the model used, turbulent viscosity, or the components of the Reynolds stress tensor $\overline{u_i u_j}$. There are separate 2D/axisymmetric and/or 3D decks for each module considered.

The modules are tested using Rocketdyn's proprietary code REACT. The code utilizes an efficient solution procedure to solve Navier-Stokes equations in a non-orthogonal body-fitted coordinate system. The differential equations are discretized over a finite-volume grid using a non-staggered variable arrangement and an efficient solution procedure based on the SIMPLE algorithm for the velocity-pressure coupling is used. The modules developed have been interfaced and tested using finite-volume, pressure-correction CFD solvers which are widely used in the CFD community. Other solvers can also be used to test these modules since they are independently structured with their own discretization scheme and solver methodology. Many of these modules have been independently tested by Professor C.P. Chen and his group at the University of Alabama at Huntsville (UAH) by interfacing them with own flow solver (MAST).

CHAPTER 1

Introduction

1.1 Background

Computational Fluid Dynamics (CFD) has been used extensively for the last decade or so in analyzing complex flow phenomenon for many industrial applications, such as combustion and turbomachinery. Most flows of practical interest are turbulent and for many of them, relatively simple prediction methods are sufficient to produce results of engineering accuracy. For others, mainly flows in complex geometry with large body forces such as curvature, rotation and separation, more complex prediction methods are required.

With advancing state-of-the-art of computer technology, the range, size and complexity of flow models being applied have increased. Users have become more sophisticated and there is a constant demand for improvement. CFD codes have adapted to this demand and many general-purpose computer codes have been developed and used. As these general purpose codes become larger, their code structure becomes sophisticated and in general this structure can be divided into three main areas;

1) Numerical algorithms which include discretization methods and solution techniques.
2) Methods of dealing with complex geometry, such as grid generation, structured or unstructured grids.
3) Physical models which include turbulence models, porosity, combustion kinetics, multi-phase flows, etc.

It seems, therefore, that the practicing engineer must have the knowledge of all these elements of the CFD program in order to successfully utilize the code. Modularization of the code structure may then become necessary in order to obtain the maximum benefits from these general-purpose CFD codes. This means developing individual modular routines for the solver and other physical models. If such modules are successful they would allow users to concentrate their talents on developing and improving physical hypothesis such as turbulence models that can be easily tested using these modules.

In general, the physics of turbulence can be captured by solving the full time-dependent Navier-Stokes equations in what is termed as Direct Numerical Simulation (DNS). However, DNS is not practical for engineering purposes mainly because it is restricted to flows at low Reynolds numbers. Large Eddy Simulations (LES) are now competitive with DNS in accuracy at an order of magnitude less cost, however, it is still expensive for routine engineering calculations. Therefore, current engineering prediction methods are based on Reynolds-averaged equations, with models for the unknown Reynolds stresses which appear as the result of time-averaging the nonlinear Navier-Stokes equations. These models fall mainly into three categories; "eddy-viscosity" models, where a relation between the Reynolds stresses and mean velocity gradients at the same point in space is sought. Algebraic stress models, where the Reynolds stresses are expressed as an algebraic relation of turbulence production and dissipation. Reynolds stress models where the exact partial differential equations for the Reynolds stresses are solved after closing the higher order terms. These transport equations account for the dependence of Reynolds stresses on the history of the flow and should perform better than the eddy-viscosity models.

1.2 Outline of the Present Study

In the present work, phenomenological, single-point turbulence models coded in a modular format are developed as self-contained code decks that can be interfaced with a number of flow solvers to analyze turbulent flows in complex 2D/axisymmetric or 3D geometry. These modules are validated using Rocketdyn's REACT code and are independently tested at UAH using own code MAST.

The models that are developed in a modular form include;

1. 2D/axisymmetric single-scale k-ε model with three options for near wall treatment that include;
   - Standard Launder and Splading wall functions.
   - Chen and Patel two-layer model.
   - Lam and Bremhorst low-Reynolds number model.

2. 2D/axisymmetric multi-scale k-ε model with the standard wall function and Chen & Patel two-layer near wall treatment.

3. 2D/axisymmetric implicit algebraic stress model (ASM) based on the original work of Rodi.

4. 2D/axisymmetric full Reynolds stress turbulence model (RSM) based on the simplified linear

second moment closure model of Launder, Reece and Rodi (LRR) second moment closure.

5. 3D standard k-ε turbulence model with wall function and two-layer near wall treatments.

6. 3D algebraic stress model (ASM).

Each model is coded as a self contained, stand alone module deck that can be interfaced with a number of CFD solvers to analyze turbulent flows in complex geometry. The user can use these modules without concern as to how they are implemented and solved. The input to the modules are the mean flow variables, boundary and geometric information which are to be provided by a mean flow solver. The output of the module are the turbulent eddy-viscosity for the eddy-viscosity models and the Reynolds stresses for the second moment closure models. Moreover, source terms which are needed for the mean flow calculations are calculated and must be passed to the main solver. The modules are tested using the finite-volume REACT code and the results compared with available experimental data.

Full details of each module are given in the next chapters. Chapter 2 discusses the theory and model equations for the two-equation k-ε model used in the 2D/axisymmetric module deck. The module is evaluated with a number of benchmark problems and detailed description of the module variable names together with the input/output structure are given in appendix A. The complete listing of the module is provided at the end of the chapter. Similarly, chapter 3 discusses the theory and model equations for the 2D/axisymmetric multi-time-scale k-ε model. The 2D/axisymmetric Algebraic stress module is presented in chapter 4 and chapter 5 discusses the 2D/axisymmetric Reynolds stress module deck. Full description of the 3D k-ε turbulence model is given in chapter 6 and chapter 7 presents a full description of the 3D algebraic stress model together with module description and code listing in the appendix. Finally in chapter 8, copies of related turbulence work that are presented or published elsewhere are attached.

CHAPTER 2

2D/Axisymmetric k-ε Turbulence Model

Table of Contents

2.1 Introduction

In this section a description of the standard k-ε turbulence model that is coded as a self contained computer program to compute turbulent flow quantities in two-dimensional planar or axisymmetric geometry is given. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in Appendix A. The module has been tested as a separate self-contained unit using the REACT code [1] and was independently tested at the University of Alabama at Huntsville (UAH) using own code (MAST).

2.2 Theory and Model Equations

The k-ε turbulence module is based on the widely used single-scale two equation k-ε turbulence model (k is the turbulent kinetic energy and ε is the energy dissipation rate). The model developed originally by Launder and Spalding [2] was successful in providing good predictions for a wide range of turbulent flows. The k and ε-equations can be derived from the transport equations for the Reynolds stresses assuming fully turbulent flow.

For low-Reynolds number flows close to solid boundaries, adjustments to the model are needed to bridge the viscous dominated sublayer region with the fully turbulent flow region. The success of the wall function method depends on the universality of the turbulent flow structure near the wall. In many complex flows, however, the flow field near the wall has to be determined accurately and the traditional wall-function method is not satisfactory. This is because the specification of all turbulence quantities in terms of the friction velocity fail at separation where the flow near the wall is no longer controlled by the wall shear stress. Patel et al [3] assessed the relative performance of various models which describe the near-wall flows and found that there are still areas of improvements needed to accurately model flow behavior near the wall.

Jones and Launder [4] extended the original k-ε model to the low-Reynolds number form which allowed the calculation to be performed all the way to the wall. Numerical difficulties of accurately resolving the large gradients close to the wall necessitates resolving the wall region with a very fine grid structure. Chen and Patel [5] introduced a method to resolve the near-wall region which combines the standard k-ε model with the one-equation model of Wolfshtein [6] near the wall. In this "two-layer" model an algebraically prescribed eddy-viscosity for the wall region is coupled to the k-ε model to describe the details of the flow in the vicinity of the wall.

Momentum and continuity equations are solved up to the wall and this reduces the physical uncertainties of near-wall turbulence and the numerical difficulties of resolving the very large gradients of turbulence parameters.

For an incompressible, steady and axisymmetric turbulent flow, the Reynolds averaged momentum and continuity equations can be expressed in a generalized form as;

$$\frac{\partial(\rho u \Phi)}{\partial x} + \frac{1}{r}\frac{\partial}{\partial r}(\rho v r \Phi) = \frac{\partial}{\partial x}\left(\Gamma_{\Phi_x}\frac{\partial \Phi}{\partial x}\right) + \frac{1}{r}\frac{\partial}{\partial x}\left(r\,\Gamma_{\Phi_r}\frac{\partial \Phi}{\partial r}\right) + S_\Phi \tag{1}$$

where Φ is the dependent variable, which stands for $\Phi = u, v, w$ for the axial, radial and tangential velocities respectively. ρ is the fluid density, Γ_{Φ_x} and Γ_{Φ_r} are exchange coefficients in x and r -directions, respectively, and S_Φ is the source term for the variable Φ.

The source terms for the dependent variable are:

- Axial direction, $\Phi = u,\ \Gamma_{\Phi_x} = 2\mu_e,\ \Gamma_{\Phi_r} = \mu_e$ and

$$S_u = -\frac{\partial P}{\partial x} + \frac{1}{r}\frac{\partial}{\partial r}\left(\mu_e r \frac{\partial v}{\partial x}\right) \tag{2}$$

where μ_e is the eddy viscosity and P is the pressure

- Radial direction, $\Phi = v,\ \Gamma_{\Phi_x} = \mu_e,\ \Gamma_{\Phi_r} = 2\mu_e$ and

$$S_v = -\frac{\partial}{\partial x}\left(\mu_e \frac{\partial u}{\partial r}\right) - 2\mu_e \frac{v}{r^2} + \frac{\rho w^2}{r} - \frac{\partial P}{\partial r} \tag{3}$$

- Tangential direction, $\Phi = w,\ \Gamma_{\Phi_x} = \mu e,\ \Gamma_{\Phi_r} = \mu e$ and

$$S_w = -\frac{\rho v w}{r} - \frac{w}{r^2}\frac{\partial}{\partial r}(r\mu_e) \tag{4}$$

Equations 2, 3, and 4 above are the momentum equations that are solved by the CFD solvers. However, in order to close the equations and determine the eddy viscosity different turbulence models are used.

The present module utilizes the k-ε model. In this model two equations for the turbulent kinetic energy k and its dissipation ε which have the same general form as equation (1) are solved.

For the turbulent kinetic energy equation

$$\Phi = k,\ \Gamma_{\Phi_x} = \Gamma_{\Phi_r} = \mu + \frac{\mu_t}{\sigma_k} \text{ and } S_\Phi = G - \rho\varepsilon \tag{5}$$

For the energy dissipation equation

$$\Phi = \varepsilon,\ \Gamma_{\Phi_x} = \Gamma_{\Phi_r} = \mu + \frac{\mu_t}{\sigma_\varepsilon} \text{ and } S_\Phi = \frac{\varepsilon}{k}(C_1 f_1 G - C_2 f_2 \rho\varepsilon) \tag{6}$$

where σ_k and σ_ε are turbulent Prandtl/Schmidt numbers for k and ε respectively, and G denotes the rate of production of the turbulent kinetic energy and is expressed as:

$$G = \mu_e \left\{ 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2 + \left(\frac{v}{r}\right)^2\right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial r} - \frac{w}{r}\right)^2 \right\} \tag{7}$$

where μ is the dynamic viscosity, and μ_t is the turbulent viscosity,

$$\mu_t = C_\mu f_\mu \rho \frac{k^2}{\varepsilon} \tag{8}$$

and $\mu_e = \mu + \mu_t$

C_μ, C_1, C_2, σ_k and σ_ε are constants whose values are 0.09, 1.44, 1.92, 1.0, 1.0, respectively and f_1, f_2 and f_m are damping functions.

Near a wall, turbulent flow can be divided into two regions, the inner viscous sublayer where low turbulent Reynolds number effects are important and the velocities decrease rapidly to zero at the wall, and the outer fully turbulent region. The successful application of the k-ε turbulence model for many complex flows depends to a large extent on how accurately the flow field near the wall is determined. In the present module three different models are used to treat this thin sublayer region, they include;

Wall function method, where

$$u^+ = y^+ \qquad at\ y^+ \leq 11.6 \tag{9}$$

$$u^+ = \frac{1}{\kappa} \ln(Ey^+) \ at\ y^+ \geq 11.6 \tag{10}$$

where, $u^+ = \frac{u}{u_\tau}$, $y^+ = \frac{u_\tau y}{\nu}$ and $u_\tau = \sqrt{\tau_w/\rho}$

τ_w is the wall shear stress which can be determined from

$$\tau_w = \frac{\mu u_p}{\delta} \ for\ y^+ \leq 11.6 \tag{11}$$

$$\tau_w = \frac{\kappa C_\mu^{0.25} \rho u_p k^{0.5}}{\ln[E C_\mu^{0.25} \rho \delta k^{0.5}/\mu]} \ for\ y^+ > 11.6 \tag{12}$$

Here, u_p denotes the velocity component parallel to the wall at the first grid point p from the wall.

δ is the normal distance from the wall and κ is a constant = 0.42.

In this approach, k and ε equations are solved with $f_\mu = f_1 = f_2 = 1$, only in the fully turbulent region beyond some distance from the wall. Boundary conditions i.e., velocity components and turbulent parameters at that distance are specified in terms of the friction velocity u_τ.

In the low-Reynolds number model, the flow is resolved all the way to the wall with a very fine mesh. Many models have been proposed that are based on the k-ε model and differ mainly in the choice of the damping functions f_μ, f_1 and f_2 to bridge the gap between the sublayer and the fully turbulent region. The model due to Lam & Bremhorst [7] is used in this work, where;

$$f_\mu = [1 - \exp(-0.016 R_y)]^{1/2} (1 + \frac{20.5}{R_t})$$

$$f_1 = 1 + \left(\frac{0.06}{f_\mu}\right)^3 \quad and \quad f_2 = 1 - exp\,(-R_t^2)$$

where, $R_y = \frac{k^{1/2}y}{\nu}$ and $R_t = \frac{k^2}{\nu\,\varepsilon}$ are turbulent Reynolds number.

These damping functions tend to unity with increasing distance from the wall.

In the two-layer model due to Chen and Patel [5], a simple algebraically prescribed eddy-viscosity model for the wall region is coupled to the k-ε model for the outer flow to describe the flow details. Unlike the low-Reynolds number model that requires the solution of transport equations for both k and ε all the way to the wall, the one-equation model requires the solution of only the turbulent kinetic energy equation in the sublayer region while algebraically specifying the eddy viscosity and energy dissipation.

$$\nu_t = C_\mu \frac{k^{1/2}}{L_\mu} \quad and \quad \varepsilon = \frac{k^{3/2}}{L_\varepsilon}.$$

The length scales L_μ and L_ε contain the necessary damping effects in the near-wall region in terms of the turbulence Reynolds number R_y.

$$L_\mu = C_l\, y\, [1 - exp\,(-R_y / A_\mu)] \tag{13}$$

$$L_\varepsilon = C_l\, y\, [1 - exp\,(-R_y / A_\varepsilon)] \tag{14}$$

L_μ and L_ε become linear and approach $C_l y$ with increasing distance from the wall. $C_l = \kappa\, C_\mu^{-0.75}$ and $A_\varepsilon = 2C_l$. Chen and Patel [5] used $A_\mu = 70$.

The damping effects decay rapidly with distance from the wall independent of the magnitude of the wall shear stress. The matching between the one-equation and the standard k-ε models is carried along prescribed grid lines where $R_y \sim 200$.

For flows in rotating ducts a modification was made by Chen and Guo [8] to reflect the effects of a system rotation on the length scales L_μ and L_ε, as;

$$L_\mu = L_{\mu 0} \, [\, 1.0 + 1.3 \, (\, 0.4 \frac{\partial U}{\partial y} - 0.8 \, \Omega \,) \, \Omega \, (\frac{k}{\varepsilon})^2 \,]^{1.5}$$

$$L_\varepsilon = L_{\varepsilon 0} \, [\, 1.0 + 1.3 \, (\, 0.4 \frac{\partial U}{\partial y} - 0.8 \, \Omega \,) \, \Omega \, (\frac{k}{\varepsilon})^2 \,]^{0.5}$$

Moreover, the function f_2 in the dissipation equation is modified to

$$f_2 = f_2 + Ri$$

where Ri is a Richardson number to reflect the effects of streamline curvature due to rotation and is defined as

$$R_i = (\, 0.4 \, \omega_k - 0.8 \, \Omega_k \,) \, \Omega_k \, (\frac{k}{\varepsilon})^2$$

where $\omega_k = \epsilon_{ijk} \frac{\partial U_i}{\partial x_j}$ is the local mean vorticity.

The above modification to account for streamline curvature and rotation seemed adequate in the framework of two equation k-ε modeling. Other modifications have also been considered but not implemented in this module and can be referred to in Hadid and Sindir [9].

2.3 Module Evaluation

The single scale k-ε turbulent module was evaluated by comparison with published experimental data. One of the test problems considered is the two dimensional incompressible turbulent flow over a backward facing step with and without rotation (see figure 1) to compare with the experiment of Rothe and Johnston[10]. While the mean flow is in the x -y plane, the channel is rotated with constant angular velocity Ω about the z -axis. The ratio of the channel width to the step height is very large so that the secondary flow can be ignored, which made the flow remain two dimensional. The channel height to step ratio was set to 2 and the inlet channel height (h) equals to the step height (H). The Reynolds number based on the uniform inlet velocity was about 5500. The rotation number ($Ro = \Omega h/U$) was varied between $+0.06$ and -0.06.

The streamline patterns for the three different rotation numbers $Ro = -0.06, 0.0, +0.06$ by using the three different wall treatments are shown in figures 2-4. In each figure, the upper (a) and lower (c) parts correspond to $Ro = +0.06$ and $Ro = -0.06$ respectively. While the middle part (b) is the non-rotating case. It is observed that the streamline patterns are influenced by the system rotation. Suction side step extends the recirculation zone and the pressure side step reduces the recirculation zone. The reattachment length for $Ro = -0.06$ using the wall functions is larger compared to the experimental results. This is due to the fact that no Coriolis effect is accounted for in the law of the wall. The predicted variation of reattachment length with Ro (figure 5) shows reasonable correlation with the experimental data of Rothe and Johnston [10].

The single scale k-ε model using three different wall treatments with rotational stress generation terms embodied seems to capture the main effects of system rotation on turbulence structure, i.e. the suppression of turbulence level with clockwise rotation and enhancement of turbulence level with counterclockwise rotation The effects are also noticeable in the corresponding increase in the reattachment length with clockwise rotation and its decrease with counterclockwise rotation.

The other two test cases were those of Daily and Nece [11] where rotating disk cavity circulation and secondary flows are induced by a rotating wall, and Roback and Johnson [12] for a confined double concentric jets with a sudden expansion. Flow swirl in this case is induced by imposing a tangential velocity component at the outer jet. Figure (6) shows the two-dimensional axisymmetric rotating lid cavity of Daily and Nece. The flow is bounded by a disk (rotor) and a stationary end wall (stator) of a chamber. The ratio of the axial clearance between the rotor and the stator (s) to the radius of the disk (a) is 0.0255. The disk rotates with a rotational Reynolds number $R=4.4x10^6$ defined as $R = \Omega a^2/\nu$, where Ω is the disk rotational speed and ν is the kinematic viscosity.

Computations were performed on a $33x75$ grid with different grid clustering near the walls for the different near-wall models. Figure (7) shows the velocity vectors at the top region of the cavity using the wall function model. Centrifugal forces move the fluid radially outward on the disk, axially away from the disk on the wall casing, and radially inwards on the stationary end wall. Figure 8, shows the axial variations of the radial velocity component at a radial position $r/a=0.765$. The agreement is fair with some discrepancy for all near-wall models close to the rotating disk. Figure (9), shows the axial variation of the tangential velocity component at the radial position $r/a=0.765$. At the rotating disk $(x=0)$, the tangential velocity approaches the value ($a\Omega$). The two-layer near wall model seem to offer closer agreement with the data than the other two models. The presence of corner regions presents a difficulty in defining the normal distances used in the definition of turbulent Reynolds number. In the present analysis, values of the normal distance were based on the normal distance to the nearest solid boundary.

Predictions of the experiments of Roback and Johnson [12] have been presented by several workers, e.g. Sloan et al. [13] and Durst and Wennerberg [14]. Unfortunately, inlet flow profiles were not provided in the experiment. Therefore, the present calculations were started at the expansion plane using the measured velocity profile at *5 mm* downstream of the expansion after some adjustments near the edges of the coaxial jets. Measurements of main turbulent intensities were used to calculate inlet values of the turbulent kinetic energy. Energy dissipation rate was estimated from $\varepsilon = C_\mu k^{3/2}/L$, where L is a length scale of turbulence at the inlet of the order of $10^{-4}\ m$.

Figure 10, shows an illustration of the test chamber geometry. The chamber diameter is about twice the secondary tube diameter. The exit from the 8-bladed, 30^o, free vortex swirl generator is located approximately *0.005 m* upstream from the confluence plane.

A prominent phenomenon in axisymmetric swirling flows in such geometry is the "bubble" or vortex breakdown which has been studied extensively [15-18]. In the present numerical simulation of the experiment, a *150x100* grid nodes was used with different clustering on the walls for the different near-wall models used. Figure 11, shows the velocity vectors indicating the presence of a closed recirculation zone at the center with additional zones at the corner downstream and between the inner jet and the outward diverted secondary jet. The figure also shows flow diversion outwards with high gradients characterized by large turbulent shear and fluctuation levels. Comparisons were made of the radial variations of flow variables at two axial locations, $x=0.025m$ upstream of the vortex bubble and $x=0.102m$ located inside the bubble. Figure (12), shows the radial variation of the axial velocity profile at $x=0.025m$ using the wall function, two-layer and the low Reynolds number models. Fair agreement by the different models is shown. They also seem to predict small negative velocities at a radial position $r\sim0.0153m$ (the interface between the inner and outer jets), slightly under predicted in strength and width. Figure (13) shows the radial variation of the axial velocity profiles at $x=0.102m$. The two-layer model shows a better agreement with the experimental data.

Radial variations of the tangential velocity at $x=0.025m$ is shown in figures 14. The figure shows that the two-layer model offers better agreement with the experiment as compared with the wall function or the low Reynolds number models.

In general, the calculations shown above indicate that the two layer model seem to offer a better comparisons with the experimental results. The three near-wall models are built in the standard two-dimensional/axisymmetric k-ε turbulence module. The structure of the module will be discussed next together with the details of interfacing with a flow solver and descriptions of variables.

REFERENCES

1. Chon, J., Hadid, A. and Hamakiotes C. "REACT-2D Version 2.6 User's Manual", CFD Technology Center, Rocketdyne Division/Rockwell International, 1989

2. Launder, B. E and Spalding, D. B. "The Numerical Computation of Turbulent Flows", Computer Methods in Applied Mechanics and Engineering, vol. 3, pp. 269-289, 1974.

3. Patel, V. C., Rodi, W. and Scheuerer, G. "Turbulence Models for Near-Wall and Low-Reynolds Number Flows: A Review", AIAA Journal, vol. 23, no. 9, pp. 1308-1319, 1985.

4. Jones, W. P. and Launder, B. E. "The Calculation of Low-Reynolds Number Phenomena With a Two-Equation Model of Turbulence", Int. J. Heat and Mass Transfer, vol. 16, pp. 1119-1130, 1973.

5. Chen, H. C. and Patel, V. C. "Near-Wall Turbulence Models for Complex Flows Including Separation", AIAA Journal, vol. 26, no. 6, pp. 641-648, 1988.

6. Wolfshtein, M. "The Velocity and Temperature Distribution in One-Dimensional Flow With Turbulence Augmentation and Pressure Gradients", Int. J. Heat and Mass Transfer, vol. 12, pp. 301-318, 1969.

7. Lam, C. and Bremhorst, K. "A Modified Form of the k-ε Model for Predicting Wall Turbulence", Trans. ASME J. Fluids Eng., vol. 103, pp. 456-460, 1981.

8. Chen, C. P. and Guo, K. L. " Applications of a Two-Layer Near Wall Model to Fully Developed and Rotation Channel Turbulent Flows", NASA Contract Report, 1991.

9. Hadid, A. H. and Sindir, M. M. "Comparative Study of Advanced Turbulence Models for Turbomachinery" NASA CP-3174, 1992.

10 Rothe, P. H. and Johnston, J. P. "Free Shear Layer Behavior in Rotating Systems" Trans. ASME J. Fluids Eng., vol. 103, pp. 456-460, 1981.

11. Daily, J. W. and Nece, R. E. "Chamber Dimension Effects on Induced Flow and Frictional

Resistance of Enclosed Rotating Disks" Trans. ASME J. Basic Eng., pp. 217-232, 1960.

12. Roback, R. and Johnson, B. "Mass and Momentum Turbulent Transport Experiment With Confined Swirling Co-Axial Jets: NASA CR-168252, 1983.

13. Sloan, D. G., Smith, P. J. and Smoot, L. D. "Modeling Swirl in Turbulent Flow Systems" Prog. Energy Combust. Sci., vol. 12, pp. 163-250, 1986.

14. Durst, F. and Wennerberg, D. "Numerical Aspects of Calculations of Confined Swirling Flows With Internal Recirculations" Int. J. for Numerical Methods in Fluids, vol. 12, pp. 203-224, 1991.

15. So, K. L. "Vortex Phenomena in Conical Diffuser" AIAA J., vol5, pp. 1072-1078, 1967.

16. Garg, A. K. and Leibovich, S. "Spectral Characteristics of Vortex Breakdown Flowfields" Phys. Fluids, vol. 22, pp. 2053-2064, 1979.

17. Faler, J. H and Leibovich, S. "Disrupted States of Vortex Flow and Vortex Breakdown" Phys. Fluids, vol. 20, pp. 1385-1400, 1977.

18. Escudier, M. P. and Zehnder, N. "Vortex Flow Regimes" J. Fluid Mech., vol. 115, pp. 105-121, 1982.

19. Stone, H. "Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations", SIAM J. Num. Anal., vol. 5, pp 530 - , 1968.

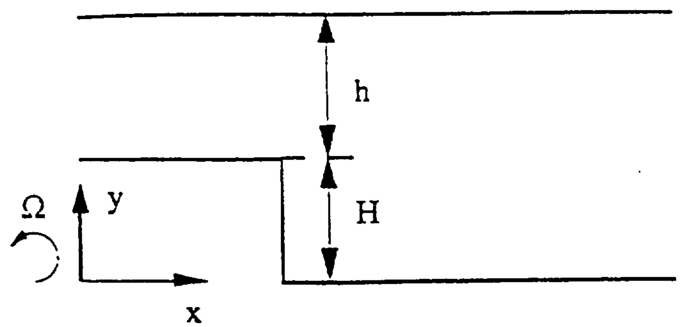
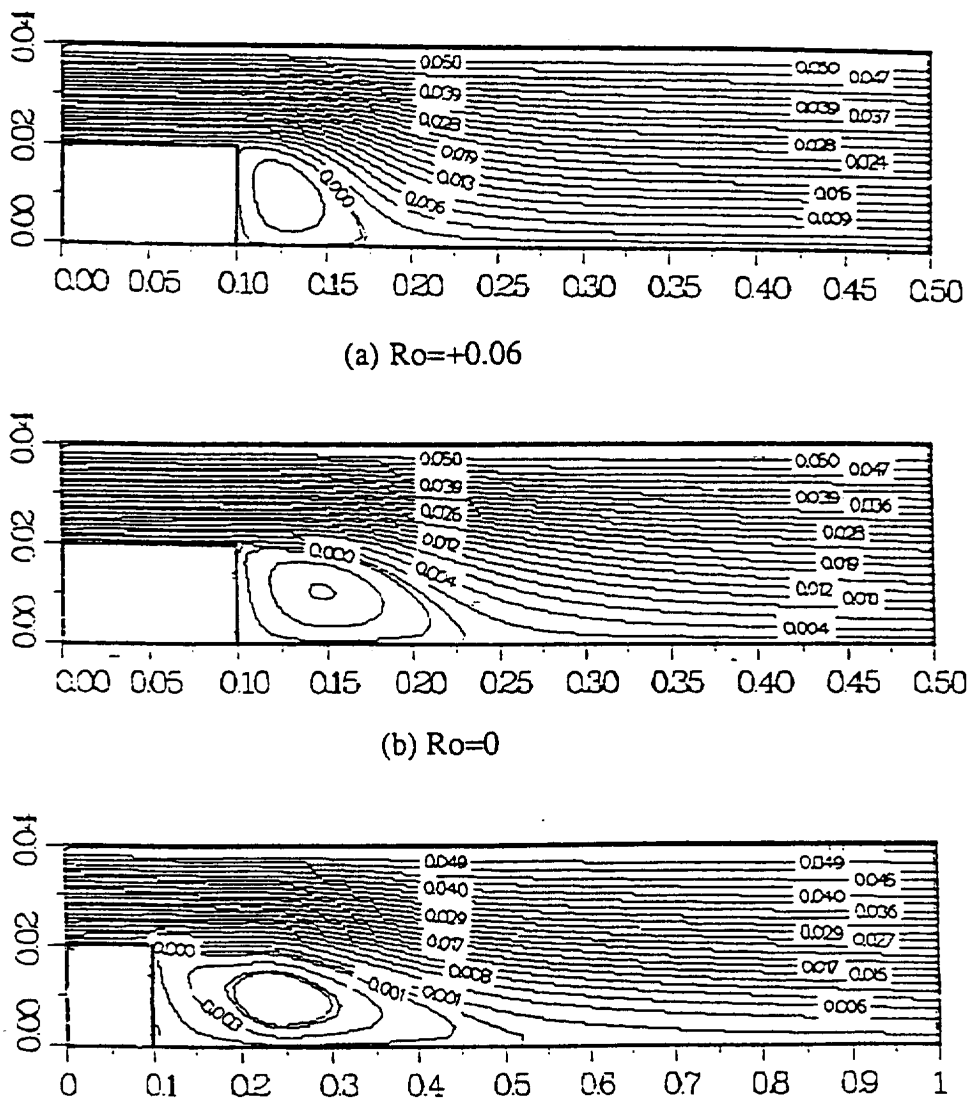


Figure 1. **Rotating backward facing step**

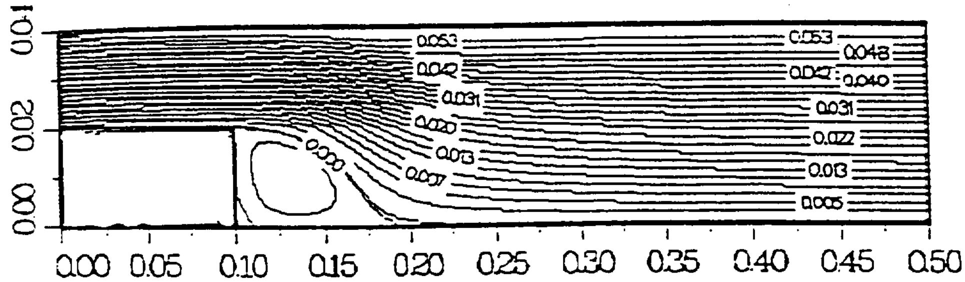


(a) Ro=+0.06

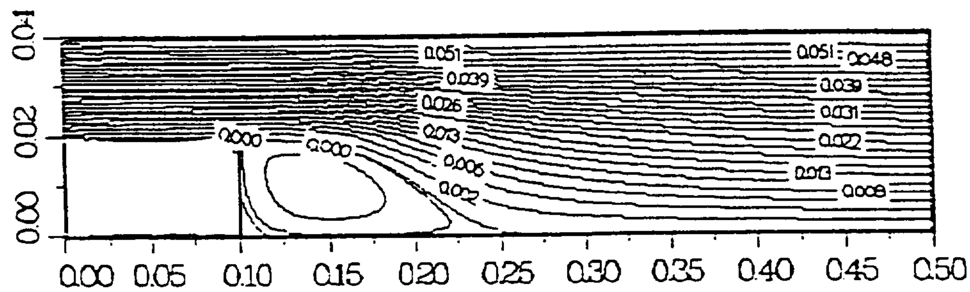
(b) Ro=0

(c) Ro=-0.06

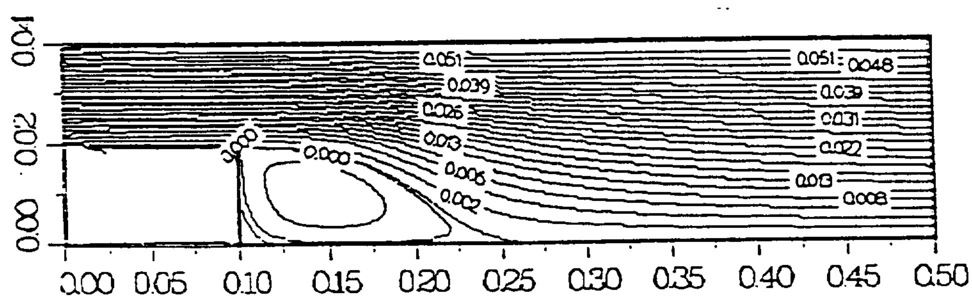
Figure 2. **Stream-function contours using wall function near wall treatment**



(a) Ro=+0.06

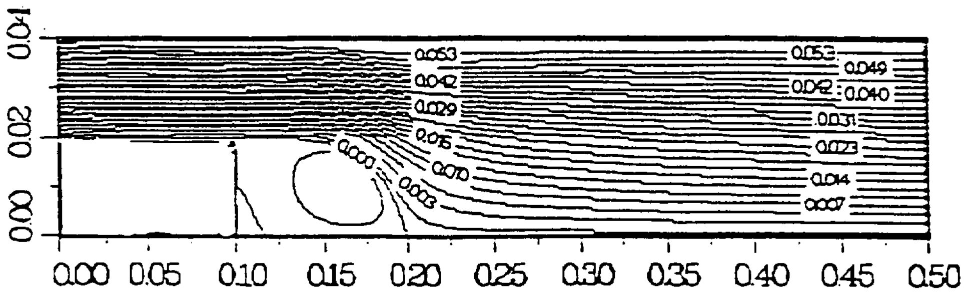


(b) Ro=0

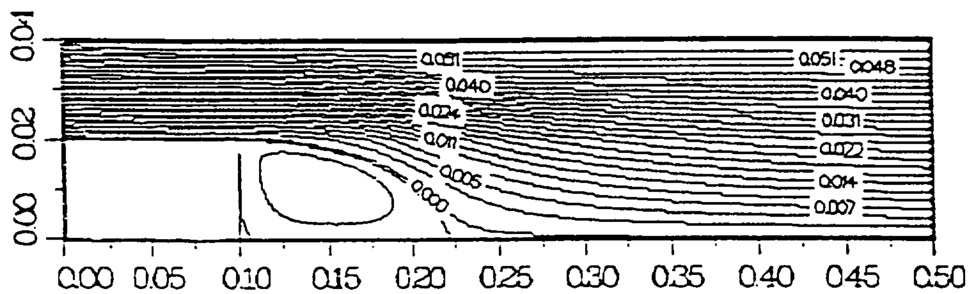


(c) Ro=-0.06

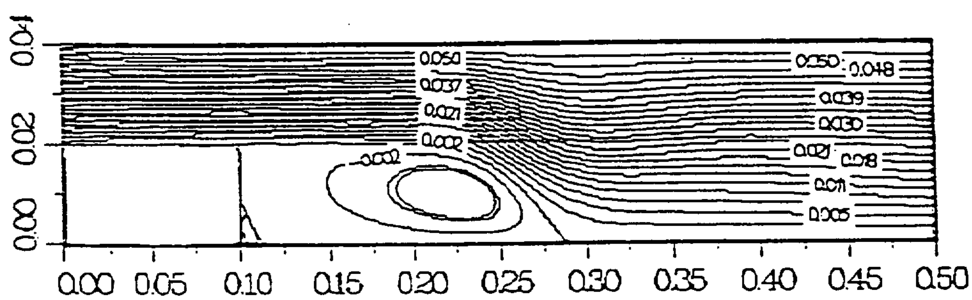
Figure 3. **Stream-function contours using the two-layer near-wall treatment**



(a) Ro=+0.06



(b) Ro=0



(c) Ro=-0.06

Figure 4. **Stream-function contours using the low-Reynolds number near-wall treatment**

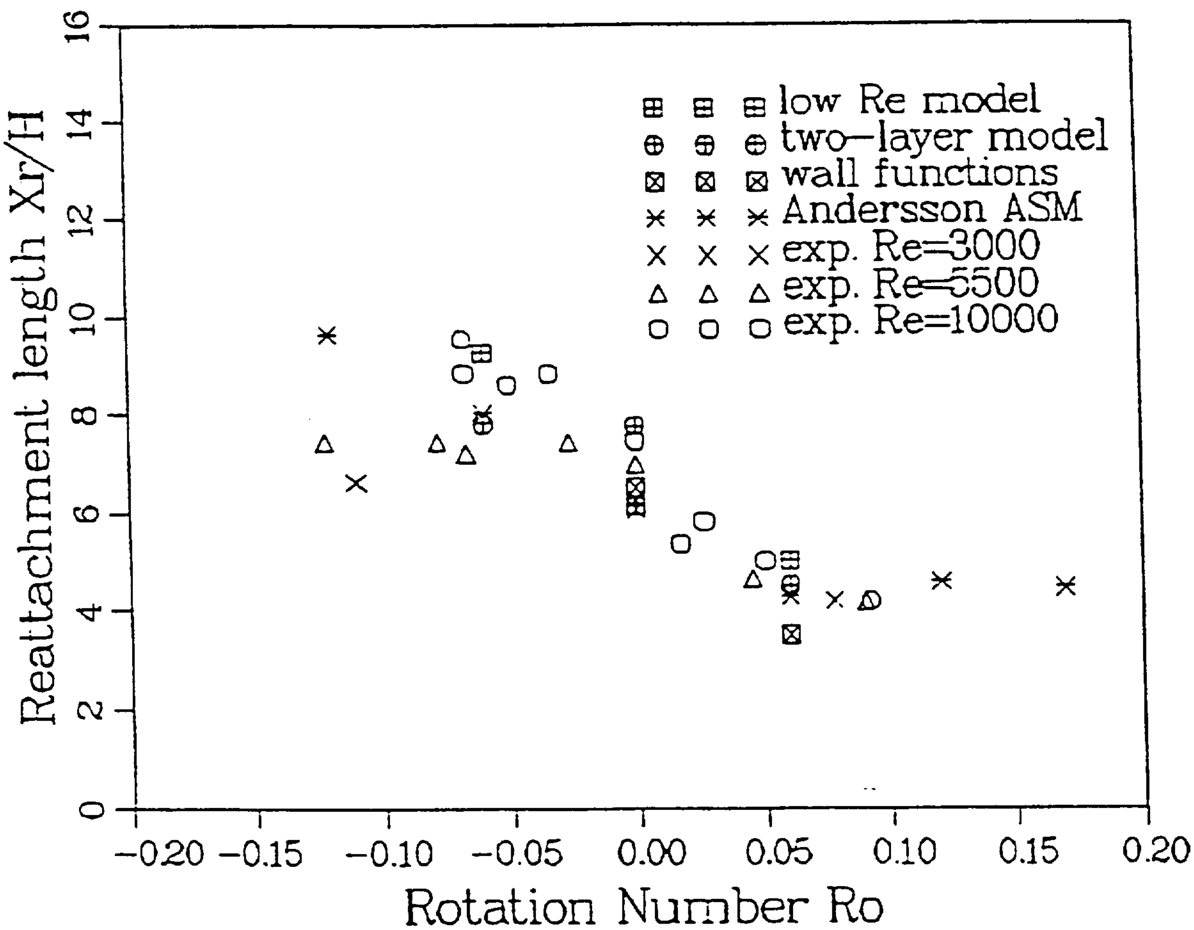


Figure 5. **Reattachment length as a function of the rotation number**

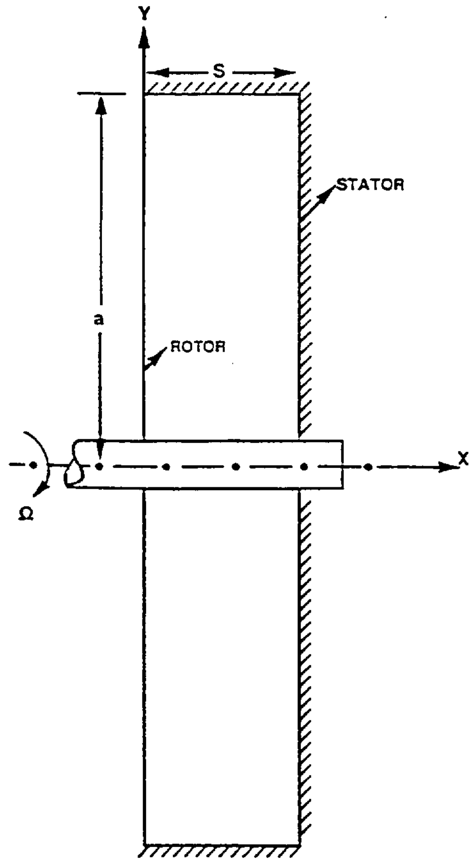


Figure 6. Rotating lid cavity

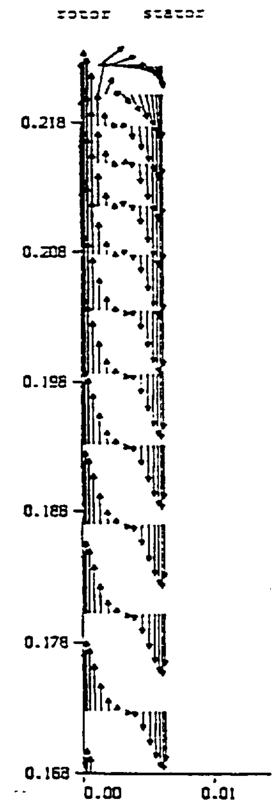


Figure 7. Velocity vectors (Re = 4.4x10^{6})

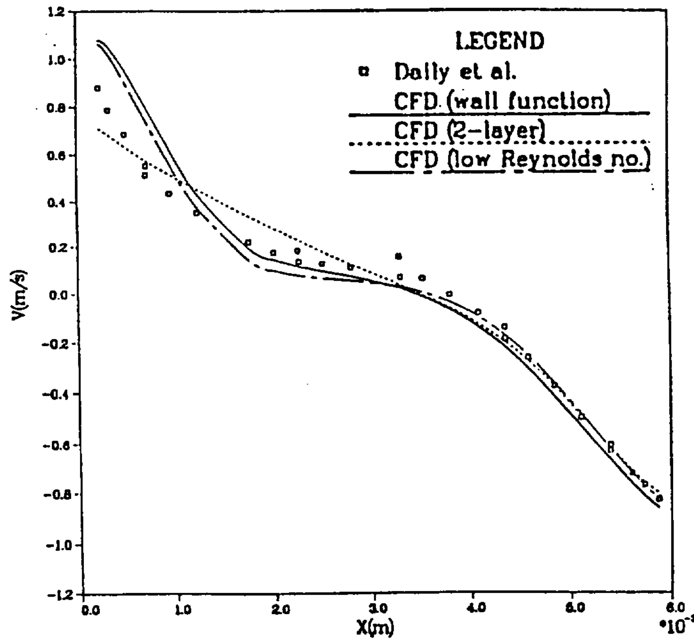


Figure 8. Axial distribution of radial velocity at $r/a = 0.765$

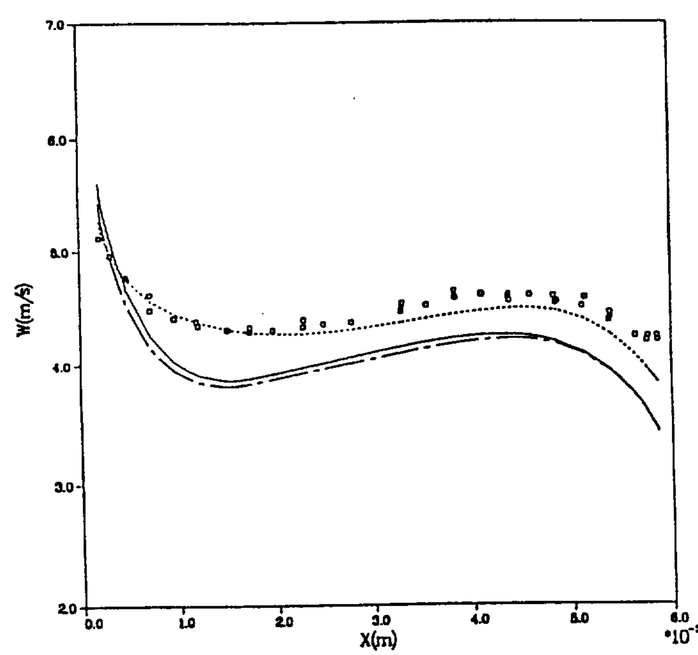


Figure 9. Axial distribution of tangential velocity at $r/a = 0.765$

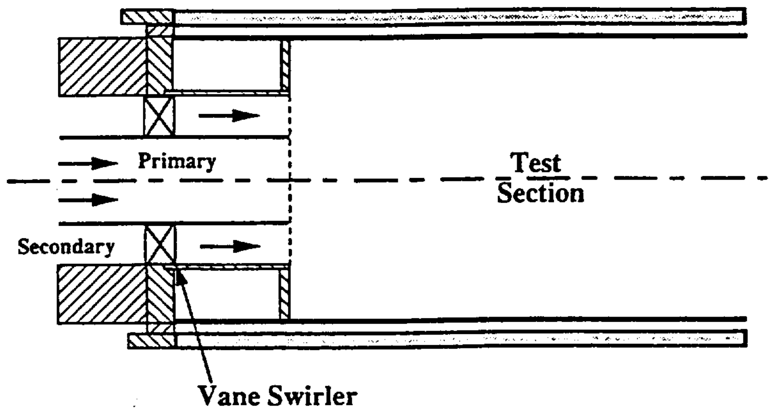


Figure 10. Roback & Johnson's swirling coaxial jets discharging into an expanded duct

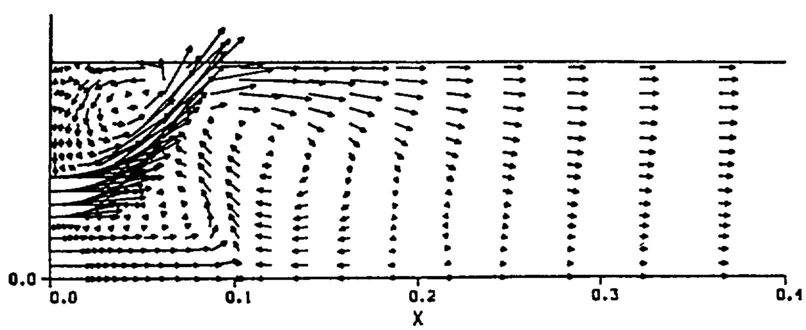


Figure 11. Velocity vectors

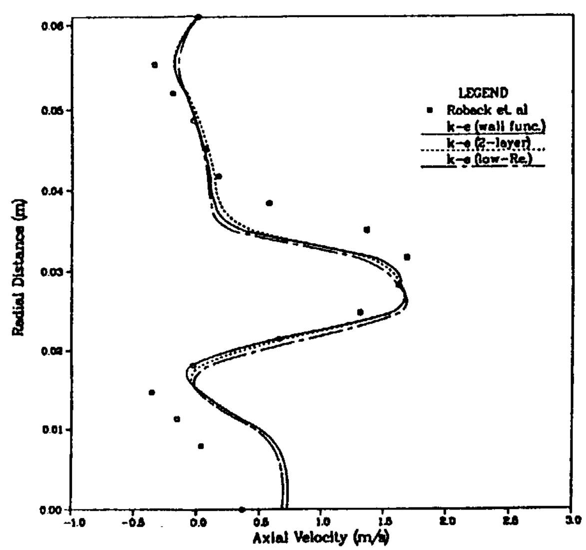


Figure 12. Radial profile of axial velocity at $x = 0.025$m

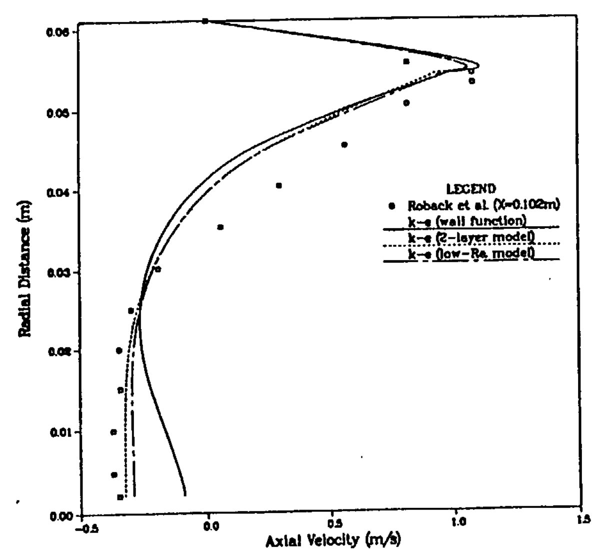


Figure 13. Radial profile of axial velocity at $x = 0.102$m

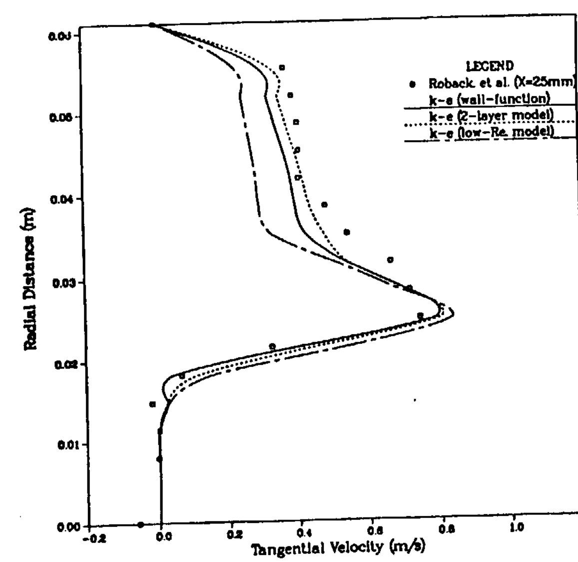


Figure 14. Radial profile of tangential velocity at x = 0.025m

APPENDIX A

2D/Axisymmetric k-ε Turbulence Module Deck

A.1 Introduction

In an attempt to modularize the k-ε turbulent physical model -a difficult task as many CFD users may know. A self-contained, stand-alone turbulence module has been constructed that computes turbulent flow quantities using the standard k-ε turbulence model. The module is structured to be flexible with options for three near-wall treatments. It can be easily accessed by the user and interfaced with own CFD solvers to calculate turbulent flows.

It is hoped that the program is sufficiently "full proof" and user friendly. However, care must be exercised to identify the limitations of the module to be compatible with the flow solver. Module capabilities and input/output structure is described next in details followed by a FORTRAN listing of the module.

A.2 Program KEMOD

This is basically the solver for the k and ε - transport equations. It reads through its argument list different variables from the calling flow solver. These variables are described below where, each variable name ends with either an (I) for Integer variable, (R) for Real variable or (L) for Logical variable.

The flow chart of the program is shown in Figure A.1. It shows the main operations performed by the code.

<u>List of Argument Variable Names</u>

NIMI	Number of cell nodes in the I- or ξ-coordinate lines. (input from flow solver)
NJMI	Number of cell nodes in the J- or η-coordinate lines. (input from flow solver)

XR	Grid node locations in the x or ξ-direction, dimensioned to XR (NX,NY) (input from flow solver)
YR	Grid node locations in the y or η-direction, dimensioned to YR (NX,NY) (input from flow solver)
UR	Axial or x-direction velocity (u), dimensioned as UR (NX,NY) (input from flow solver
VR	Radial or y-direction velocity (v), also dimensional as VR (NX,NY) (input from flow solver)
WR	Azimuthal velocity (w), dimensional WR (NX,NY) (input from flow solver)
TER	Turbulence kinetic energy k, dimensioned TER (NX,NY) (calculated in KEMOD and returned to flow solver)
EDR	Turbulent energy dissipation rate ε, dimensioned EDR (NX,NY) (calculated in KEMOD and returned to flow solver)
URFKR	Under-relaxation factor for k -equation (input from flow solver)
URFER	Under-relaxation factor for ε -equation (input from flow solver)
PRTKR	Prandtl/Schmidt number for turbulent energy-equation, assumed known (input from flow solver)
PRTER	Prandtl/Schmidt number for turbulent energy dissipation equation, assumed known (input from flow solver)
GR	= 1.0 if second order upwinding is desired = 0.0 if first order upwinding is used (input from flow solver. Usually calculation of k and ε are not very sensitive to the order of upwinding used)

F1R Mass flux variable at cell faces in x- or ξ-direction, dimensioned F1R (NX,NY) (input from flow solver)

F2R Mass flux variable at cell faces in y or η-direction, dimensioned F2R (NX,NY) (input from flow solver)

ITERI Iteration number (input from flow solver)

VISCOSR Dynamic viscosity (input from flow solver)

VISR Eddy viscosity, dimensioned VISR (NX,NY) (calculated in KEMOD and returned to main solver)

AKSIL Logical variable for axisymmetric geometry (AKSIL=·TRUE·) or plain geometry (AKSIL=·FALSE·) (input from flow solver)

LREL Logical variable for Lam & Bremhorst's low-Reynolds number model (LREL=·TRUE·) or others (LREL=·FALSE·) (input from flow solver)

LAY2L Logical variable for Patel's two-layer model if (LAY2L=·TRUE·) or others (LAY2L = ·FALSE·) (input from flow solver)

C1R Turbulence model constant, C_1 (input from flow solver)

C2R Turbulence model constant, C_2 (input from flow solver)

CMUR Turbulence model constant, C_μ (input from flow solver)

I2LWI Grid line location from the west wall in the x-direction for the two-layer model (input from flow solver)

I2LEI Grid line location from the east wall in the x-direction for the two-layer model (input from flow solver)

J2LSI Grid line location from the south wall in the y-direction for the two-layer model (input from flow solver)

J2LNI	Grid line location from the north wall in the y-direction for the two-layer model (input from flow solver)
JTBEI	Boundary condition flag along east boundary must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set JTBEI to NJ*2, and similarly for other boundaries, dimensioned JTBEI (NY) (input from flow solver)
JTBWI	Boundary condition flag along west boundary, dimensioned JTBWI (NY) (input from flow solver)
ITBNI	Boundary condition flag along north boundary, dimensioned ITBNI (NX) (input from flow solver)
ITBSI	Boundary condition flag along south boundary, dimensioned ITBSI (NY) (input from flow solver)

Program KEMOD is interfaced with the main flow solver by a call to KEMOD with its arguments. For iterative flow solvers KEMOD is called within the iteration sequence after the solution of the momentum equations where the mean velocities are passed to KEMOD. There are different flow solvers utilizing different schemes from staggered to nonstaggered grid arrangement and for nonorthogonal coordinate system there are at least three alternatives to the choice of the velocity components

i. Cartesian velocity components
ii. Contravariant velocity components
iii. Covariant velocity components

The Cartesian velocity components are the most widely used and have the advantage of simple formulation of the governing equations. Whatever the arrangement used, mass fluxes at cell faces are required and passed to KEMOD as F1R and F2R in both directions. The location of other variables such as k and ε are at the cell center or cell nodes.

The module starts by reassigning variable names passed to it from flow solver to names that are shared with the different subroutines of the module in a common statement file "KEMOD·COMMON". Then a check is made if it is the first iteration in which case the grid file "GRIDF" is called -after passing the grid node locations XR & YR in KEMOD- in order to calculate grid related quantities which will be explained later. The need to call GRIDG can be waived if all the grid data are passed to the module. That is all the information about the grid such as interpolation factors FX and FY, cell areas (ARE) and volumes (VOL) and normal distances of first grid point from grid boundaries (DNS from south boundary, DNN - from north boundary, DNW - from west boundary and DNE - from east boundary).

After this a call to subroutine CALCE is made to calculate the turbulent kinetic energy k (with the identifier IPHI=1) followed by a check if the low-Reynolds number model or the two-layer model are to be used in which case subroutine TWOLAY is called. The energy dissipation equation is solved next by a call to subroutine CALCE again with the identifier IPHI=2. The turbulent viscosity is updated next by calling subroutine MODVIS. A brief description of each subroutine is given next.

A.3 Subroutines

GRIDG

Before calling this subroutine, the coordinates of all grid nodes, defined in reference to a fixed Cartesian coordinate frame are read. Figure A.2 shows the position of cell and grid nodes.

This subroutine is called only once to calculate coordinates of grid nodes (intersection of grid lines) and geometrical properties of the grid (cell areas and volumes, interpolation factors, normal distances of near-boundary cell nodes from boundary). All variables including grid node coordinates are converted to one-dimensional arrays. These are formed by scanning the grid in J-direction (figure A.2) for I=1, and then repeating for all I's. The position of any node in one-dimensional array is therefore defined as;

$$IJ = (I,J) = (I-1) * NJ + J$$

The actual number of grid nodes is one row and one column less than for all cell nodes. For I = NI and J = NJ fictitious grid nodes are introduced which have the same coordinates as actual nodes on NI-1 in I-direction and NJ-1 in J-direction.

The subroutine then calculates interpolation factors which are associated with cell nodes and are used in the main program to calculate values of dependent variables at locations other than cell nodes (cell centers). Definition of these are given in Figure A.3. Cell areas and volumes are calculated next followed by calculations of normal distances of near-boundary nodes from all four outer boundaries.

CALCE (PHI, IPHI)

This subroutine solves the linearized and discretized transport equations for the turbulent energy k and the energy dissipation rate ε. The two dummy parameters in the calling statement, PHI and IPHI, represent arrays containing dependent variables for which the equation is to be solved. the subroutine sets up the convective and diffusive coefficients over the entire field. Then it calculates the source terms for either k or ε transport equations. A call is made to entry MODPHI in order to modify these sources and boundary coefficients to suit the particular problem. Moreover, a check is made if the two-layer model is selected then the energy dissipation is set algebraically in the sublayer region.

The discretized equations have the form

$$A_p \Phi_p = \sum_{i=EWNS} A_i \Phi_i + S_\Phi$$

where the coefficients A_i ($i=E,W,N,S$ see figure A.3) contain both the convective and diffusive fluxes. these equations are assembled and solved by calling subroutine SOLSIP which is based on Stone's Strongly Implicit Solver [19].

TWOLAY

This subroutine is called if the two-layer or low-Reynolds number models are used. It calculates the different coefficients needed to describe the energy dissipation and eddy viscosity. In this subroutine the normal distances used in the definition of the turbulent Reynolds number R_y at corner regions are calculated based on the normal distance nearest to the solid boundary.

SOLSIP

This subroutine solves the system of linear algebraic equations for k and ε using Stone's Implicit Procedure [19]. The array RES (IJ) is used to store residuals. The sum of absolute residuals "RESORP" calculated in the first pass through this part of the routine is used as a measure of convergence of the solution process as a whole and this value is stored in RESOR (IPHI). This variable RESOR (IPHI) is passed to the main solver and if desired can be normalized and compared with the maximum error allowed there. If necessary, inner iterations counter L and the sum of absolute residuals RESORP are printed out to monitor the rate of convergence of k and ε solution. If the ratio RSM is greater than the maximum allowed for the variable in question, SOR (IPHI), and the number of inner iterations is smaller than a prescribed maximum, NSWP (IPHI), then the routine repeats the sequence of calculating the residuals, increment vectors and updating the dependent variable.

USERM

This subroutine has different ENTRY points or sections where variables are updated and boundary conditions are set.

Section MODVIS

This section calculates effective viscosity (Eq. 8). It is called after calculating k and ε. At locations where ε is close to zero (i.e., $\leq 10^{-30}$) viscosity is set to zero. A provision is made for under relaxing changes in effective viscosity which may help to stabilize oscillations and improve convergence rate.

Section MODPHI

This section is called from CALCE subroutine and sets the boundary conditions for k and ε depending on which variable being called (IDIR = 1 for k and IDIR = 2 for ε). For the k -equation, the south boundary is checked first if it is one of four options:

(1) An inflow boundary ITBS(I) = 1, where the source term is set to accept the inlet values at J = 1 (south boundary)

(2) Outflow boundary ITBS(I) = 2, where zero gradient in y or η-direction is employed.

(3) Symmetry boundary, TBS(I) = 3, where gradients normal to symmetry plane are zero.

(4) Wall boundary, ITBS(I) = 4, where the production term GENTS(I) calculated form subroutine WALLFN in program MODIFY is added to the rest of the source term SU(IJ).

Boundary conditions for the ε-equation are similar to those of k except at the wall where they are set to appropriate values for each near wall treatment.

A.4 Program MODIFY

This program is compiled separately and is called from the u and v solver routines. It basically updates the flux source term of the discretized momentum equation due to wall shear stresses. If the u-momentum equation for example is discretized in the form

$$A_p^* u_p = \sum_{i=EWNS} A_i u_i + S_u^*$$

where P, E, W, N, S are cell nodes as shown in Figure A.3, and A_p^* and A_i's contain convective and diffusive coefficients. S_u^* is the source term containing pressure gradients and cross-derivative diffusion terms and convective terms for second-order upwinding scheme. This source term is usually linearized as $S_u^* = S_u - B_p u_p$. The term B_p is usually moved to the left hand side of the equation and modifies the diagonal coefficient $A_p = A_p^* + B_p$, and the equation can be written as

$$Ap\, u_p = \sum_{i=EWNS} A_i u_i + Su$$

Then S_u and B_p are passed to subroutine MODIFY where they are modified if a wall is present (e.g., ITBS(I) = 4 for south boundary).

For an iterative flow solver using the finite-volume methodology. A typical interface and call to the k-ε module from the main flow solver can be represented by a flow chart as shown in figure A.4.

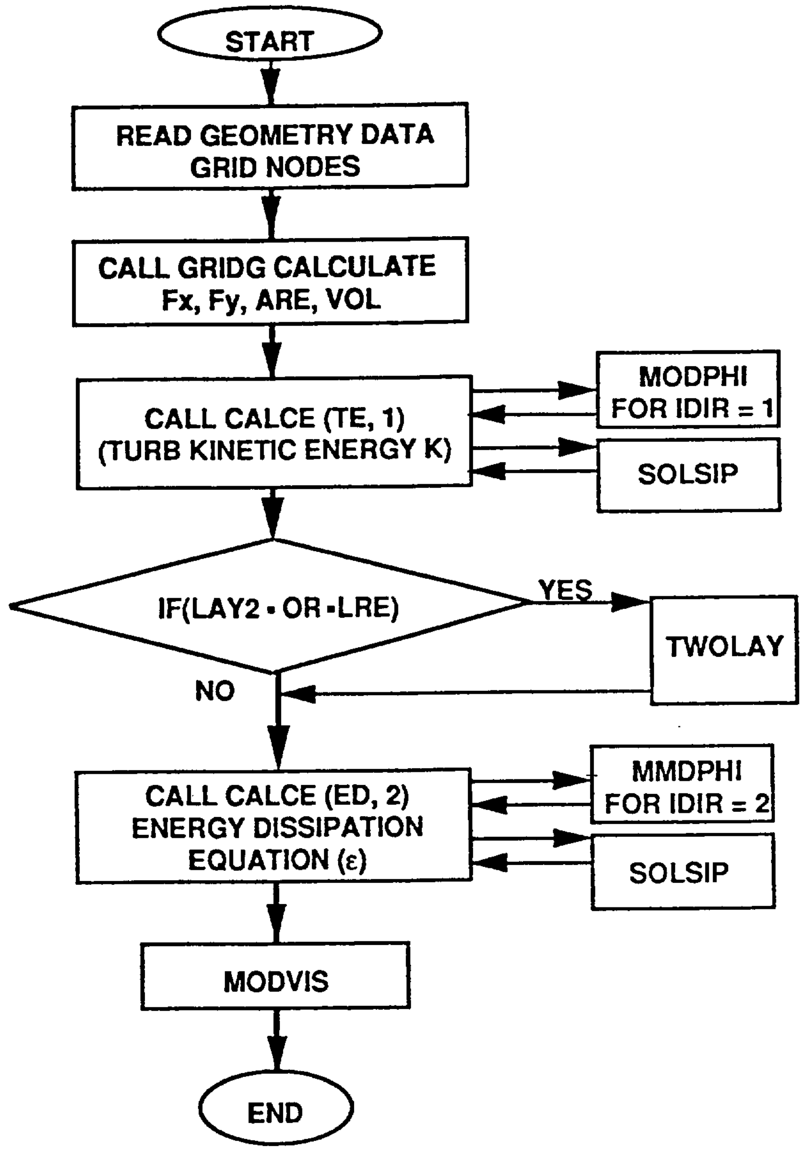


Figure A.1 2D/axisymmetric k-ε module deck flow chart

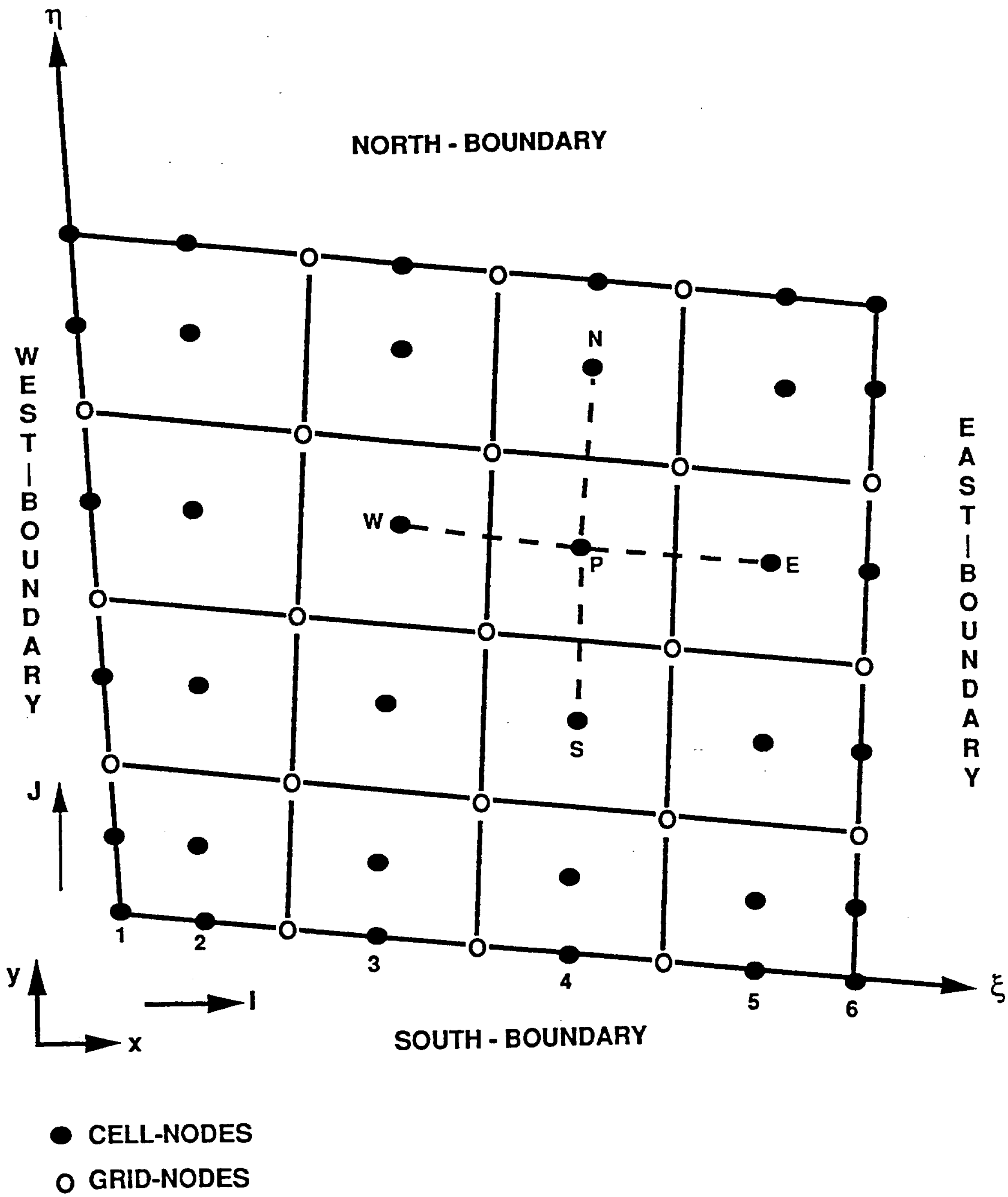
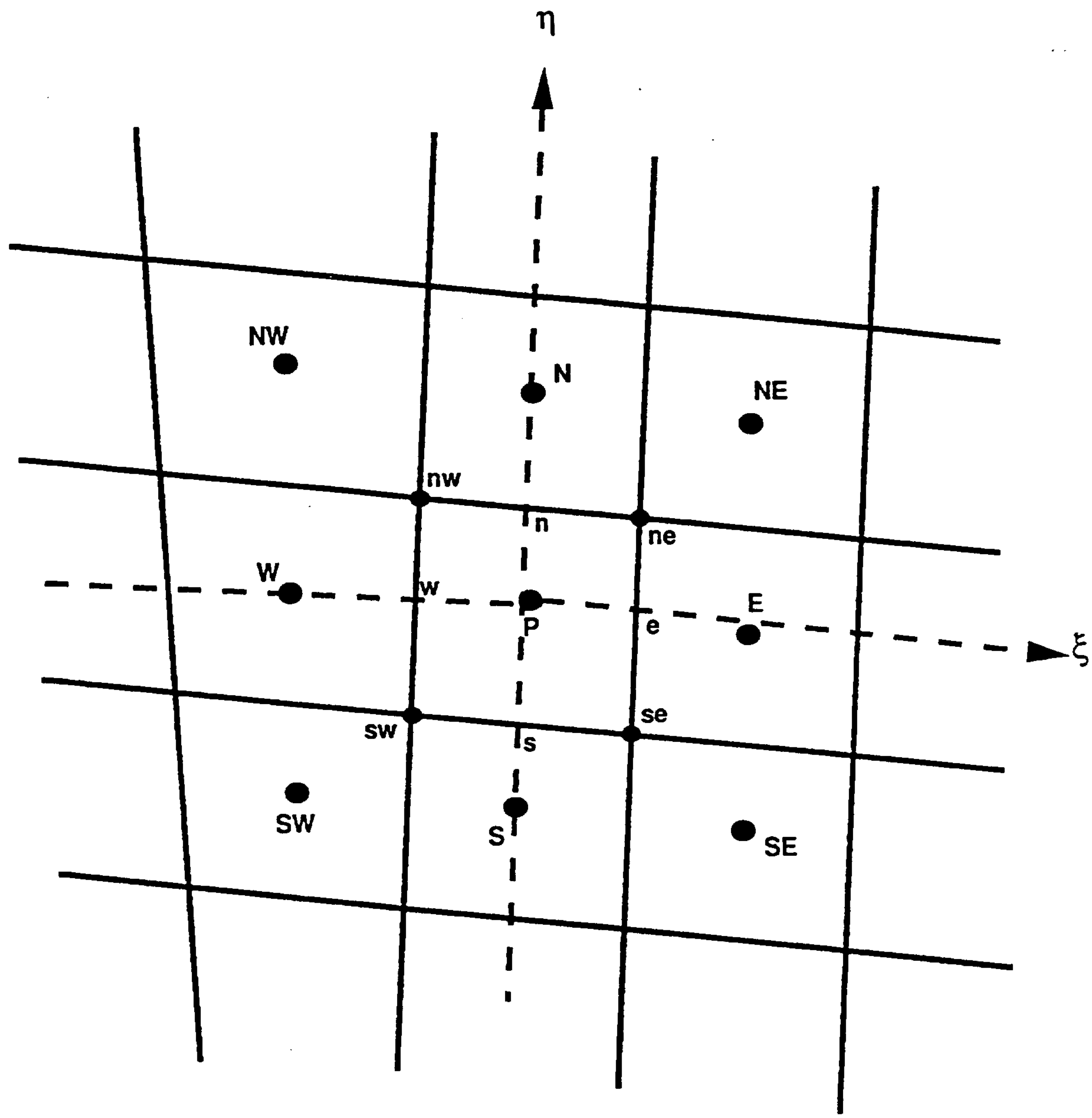


Figure A.2 Position of cell and grid nodes



$$FX_P = \frac{\overline{Pe}}{\overline{Pe} + \overline{eE}} \quad , \quad FY_P = \frac{\overline{Pn}}{\overline{Pn} + \overline{nN}}$$

Figure A.3 Definition of the interpolation factors

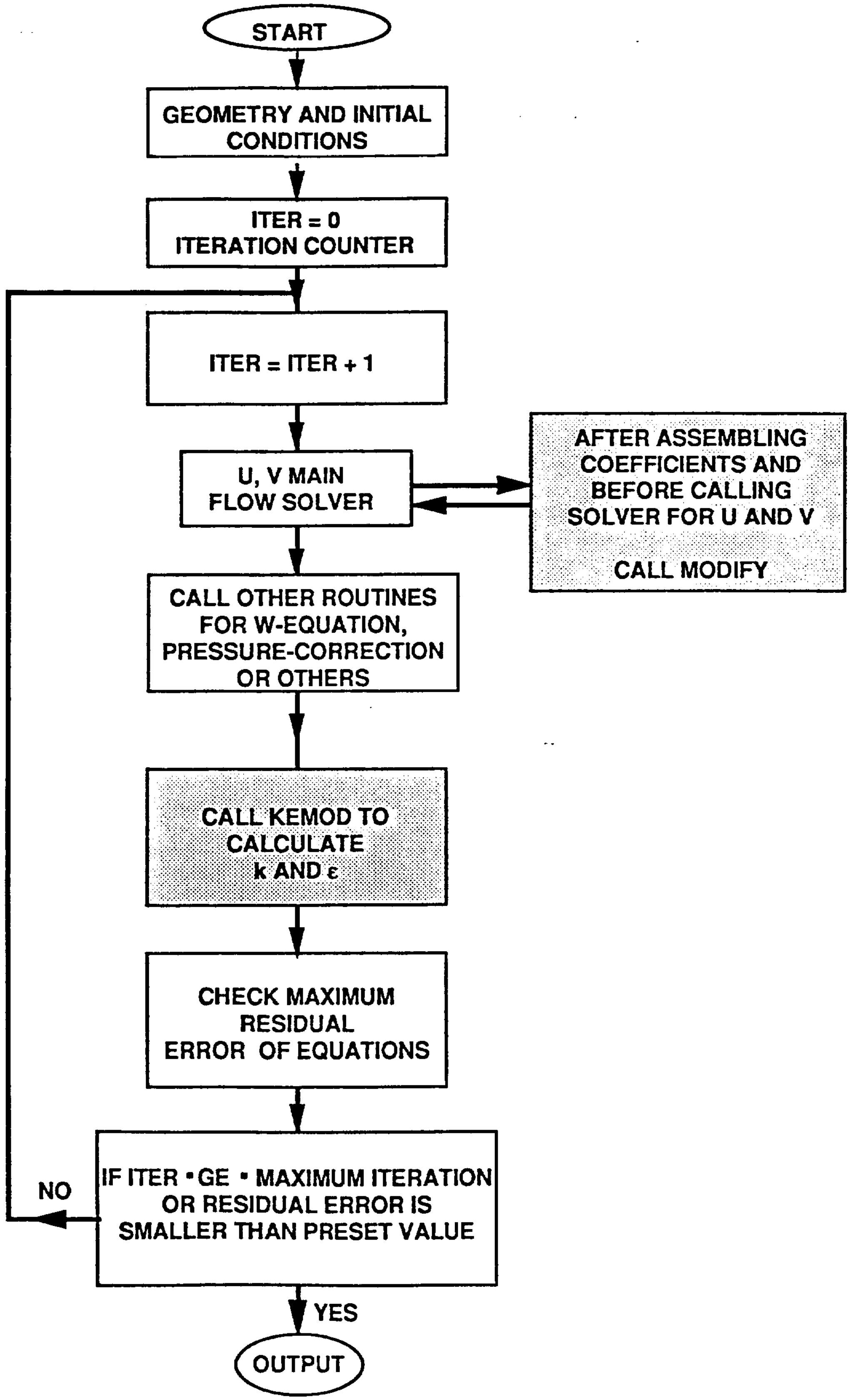


Figure A.4 Typical main flow solver with calls to the 2D/axisymmetric k-ε module

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     2D/AXISYMMETRIC SINGLE-SCALE K-E TURBULENCE MODULE             C
C                                                                     C
C     Rocketdyne CFD Technology Center                                C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  Single Scale 2-Equation with 3 Wall Treatments
C  1.) Wall Function  2.) Two Layer 3.) Low Reynolds Number
C
C
C----------------------------------------------------------
      SUBROUTINE KEMOD (NIMI,NJMI,XR,YR,UR,VR,WR,TER,EDR,
     & URFKR,URFER,PRTKR,PRTER,GR,F1R,F2R,ITERI,VISCOSR,VISR,
     & AKSIL,LREL,LAY2L,C1R,C2R,CMUR,I2LWI,I2LEI,J2LSI,
     & J2LNI,JTBEI,JTBWI,ITBNI,ITBSI)
C----------------------------------------------------------
      INCLUDE 'kemod.h'
C
      DIMENSION XR(NX,NY),YR(NX,NY),UR(NX,NY),VR(NX,NY),
     & WR(NX,NY),TER(NX,NY),EDR(NX,NY),F1R(NX,NY),F2R(NX,NY),
     & VISR(NX,NY),JTBEI(NY),JTBWI(NY),ITBNI(NX),ITBSI(NX)
C
      COMMON/GR/ X(NX,NY),Y(NX,NY)
C
      DATA GREAT,SMALL/1.E30,1.E-30/
      DATA HAF,QTR/0.5,0.25/
      DATA SOR/0.1,0.1/
      DATA NSWP/10,10/
      DATA WOMG/0.0/
C
C--EQUATE VARIABLES IN ARGUMENT LIST TO THOSE IN COMMON BLOCK
C
      NIM=NIMI
      NJM=NJMI
      NI=NIM+1
      NJ=NJM+1
      URFK=URFKR
      URFE=URFER
      PRTK=PRTKR
      PRTE=PRTER
      G=GR
      ITER=ITERI
      VISCOS=VISCOSR
      C1=C1R
      C2=C2R
      CMU=CMUR
C
C--LOGICAL VARIABLES
C
      AKSI=AKSIL
      LRE=LREL
      LAY2=LAY2L
C
C--TWO-LAYER TURBULENT PARAMETRS
C
      I2LW=I2LWI
      I2LE=I2LEI
      J2LS=J2LSI
      J2LN=J2LNI
C
      DO 10 I=1,NI
      IMNJ(I)=(I-1)*NJ
 10   CONTINUE
C
C--BOUNDARY CONDITION IDENTIFIERS
C
      DO 20 I=1,NIM
      ITBN(I)=ITBNI(I)
      ITBS(I)=ITBSI(I)
```



```
 20   CONTINUE
C
      DO 30 J=1,NJM
      JTBE(J)=JTBEI(J)
      JTBW(J)=JTBWI(J)
 30   CONTINUE
C
C
      DO 50 I=1,NI
      DO 50 J=1,NJ
         IJ=(I-1)*NJ+J
         X(I,J)=XR(I,J)
         Y(I,J)=YR(I,J)
         U(IJ)=UR(I,J)
         V(IJ)=VR(I,J)
         W(IJ)=WR(I,J)
         TE(IJ)=TER(I,J)
         ED(IJ)=EDR(I,J)
         VIS(IJ)=VISR(I,J)
         F1(IJ)=F1R(I,J)
         F2(IJ)=F2R(I,J)
 50   CONTINUE
C
C
C------------------------------------------
C---- CALCULATE GRID GEOMETRIC VARIABLES INITIALLY
C
      IF(ITER.LE.1) THEN
         CALL GRIDG
      ENDIF
C
C------------------------------------------
C
C--- CALL KINETIC ENERGY SOLVER
      CALL CALCE (TE,1)
C
C--------2-LAYER OR LOW-RE. MODELS
      IF(LAY2.OR.LRE) CALL TWOLAY
C
C--- CALL ENERGY DISSIPATION SOLVER
      CALL CALCE (ED,2)
C
C--------UPDATE AND CALCULATE EDDY VISCOSITY
      CALL MODVIS
C
      DO 60 I=1,NI
      DO 60 J=1,NJ
         IJ=(I-1)*NJ+J
         TER(I,J) = TE(IJ)
         EDR(I,J) = ED(IJ)
         VISR(I,J) = VIS(IJ)
  60  CONTINUE
C
      RETURN
      END
C
C------------------------------------------------
      SUBROUTINE GRIDG
C------------------------------------------------
C      INCLUDE 'kemod.h'
       INCLUDE 'kemod.h'
C
      COMMON/GR/ X(NX,NY),Y(NX,NY)
C
      NJ=NJM+1
      NI=NIM+1
      NINJ=NI*NJ
      DO 2 I=1,NI
      IMNJ(I)=(I-1)*NJ
 2    CONTINUE
      DO 3 I=1,NIM
```

```
      J=NJM
      X(I,J+1)=X(I,J)
      Y(I,J+1)=Y(I,J)
3     CONTINUE
      DO 4 J=1,NJ
      I=NIM
      X(I+1,J)=X(I,J)
      Y(I+1,J)=Y(I,J)
4     CONTINUE
C
C... GRID ORIGIN AT X=0, Y=0
      DO 5 I=1,NI
      DO 5 J=1,NJ
      IJ=(I-1)*NJ+J
      XX(IJ)=X(I,J)
      YY(IJ)=Y(I,J)
5     CONTINUE
C
C-----CALCULATION OF INTERPOLATION FACTORS
C
      DO 6 IJ=1,NINJ
      FX(IJ)=0.
      FY(IJ)=0.
    6 CONTINUE
C
      DO 7 J=2,NJM
      IJ=J
      FX(IJ)=0.
      LIE=NIM-1
      DO 8 I=2,LIE
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IJM=IJ-1
      IMJ=IJ-NJ
      DXP=0.5*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXPE=0.5*(XX(IPJ)-XX(IJ)+XX(IPJ-1)-XX(IJM))
      DYP=0.5*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYPE=0.5*(YY(IPJ)-YY(IJ)+YY(IPJ-1)-YY(IJM))
      DKP=SQRT(DXP**2+DYP**2)
      DKE=SQRT(DXPE**2+DYPE**2)
      FX(IJ)=DKP/(DKP+DKE)
    8 CONTINUE
      IJ=IMNJ(NIM)+J
      FX(IJ)=1.0
    7 CONTINUE
C
      DO 9 I=1,NI
      IJ=IMNJ(I)+1
      FX(IJ)=FX(IJ+1)
      IJ=IMNJ(I)+NJ
      FX(IJ)=FX(IJ-1)
    9 CONTINUE
C
      DO 10 I=2,NIM
      IJ=IMNJ(I)+1
      FY(IJ)=0.0
      LJN=NJM-1
      DO 11 J=2,LJN
      IJ=IMNJ(I)+J
      IJP=IJ+1
      IJM=IJ-1
      IMJ=IJ-NJ
      DXP=0.5*(XX(IJ)+XX(IMJ)-XX(IJM)-XX(IMJ-1))
      DXPN=0.5*(XX(IJP)+XX(IMJ+1)-XX(IJ)-XX(IMJ))
      DYP=0.5*(YY(IJ)+YY(IMJ)-YY(IJM)-YY(IMJ-1))
      DYPN=0.5*(YY(IJP)+YY(IMJ+1)-YY(IJ)-YY(IMJ))
      DEP=SQRT(DXP**2+DYP**2)
      DEPN=SQRT(DXPN**2+DYPN**2)
      FY(IJ)=DEP/(DEP+DEPN)
   11 CONTINUE
      IJ=IMNJ(I)+NJM
```



```
      FY(IJ)=1.0
   10 CONTINUE
C
      DO 12 J=1,NJ
      FY(J)=FY(J+NJ)
      IJ=IMNJ(NI)+J
      FY(IJ)=FY(IJ-NJ)
   12 CONTINUE
C
C-----CALCULATION OF CELL AREAS
C
      DO 13 IJ=1,NINJ
      ARE(IJ)=0.
   13 CONTINUE
C
      DO 14 I=2,NIM
      DO 14 J=2,NJM
      IJ=IMNJ(I)+J
      DXNESW=XX(IJ)-XX(IJ-NJ-1)
      DYNESW=YY(IJ)-YY(IJ-NJ-1)
      DXNWSE=XX(IJ-NJ)-XX(IJ-1)
      DYNWSE=YY(IJ-NJ)-YY(IJ-1)
      ARE(IJ)=0.5*ABS(DXNESW*DYNWSE-DXNWSE*DYNESW)
   14 CONTINUE
C
C-----NORMAL DISTANCE FROM THE WALL
C
      DO 15 I=1,NI
      DNS(I)=0.
      DNN(I)=0.
15    CONTINUE
C
      DO 16 J=1,NJ
      DNW(J)=0.
      DNE(J)=0.
   16 CONTINUE
C
      DO 17 I=2,NIM
      IJ=IMNJ(I)+2
      IMJ=IJ-NJ
      DXB=XX(IJ-1)-XX(IMJ-1)
      DYB=YY(IJ-1)-YY(IMJ-1)
      DXBP=0.25*(XX(IJ)-XX(IJ-1)+XX(IMJ)-XX(IMJ-1))
      DYBP=0.25*(YY(IJ)-YY(IJ-1)+YY(IMJ)-YY(IMJ-1))
      DNS(I)=DELTA(DXB,DYB,DXBP,DYBP)
      IJ=IMNJ(I)+NJM
      IMJ=IJ-NJ
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=0.25*(XX(IJ-1)-XX(IJ)+XX(IMJ-1)-XX(IMJ))
      DYBP=0.25*(YY(IJ-1)-YY(IJ)+YY(IMJ-1)-YY(IMJ))
      DNN(I)=DELTA(DXB,DYB,DXBP,DYBP)
   17 CONTINUE
C
      DO 18 J=2,NJM
      IJ=IMNJ(2)+J
      IMJ=IJ-NJ
      DXB=XX(IMJ)-XX(IMJ-1)
      DYB=YY(IMJ)-YY(IMJ-1)
      DXBP=0.25*(XX(IJ)-XX(IMJ)+XX(IJ-1)-XX(IMJ-1))
      DYBP=0.25*(YY(IJ)-YY(IMJ)+YY(IJ-1)-YY(IMJ-1))
      DNW(J)=DELTA(DXB,DYB,DXBP,DYBP)
      IJ=IMNJ(NIM)+J
      IMJ=IJ-NJ
      DXB=XX(IJ)-XX(IJ-1)
      DYB=YY(IJ)-YY(IJ-1)
      DXBP=0.25*(XX(IMJ)-XX(IJ)+XX(IMJ-1)-XX(IJ-1))
      DYBP=0.25*(YY(IMJ)-YY(IJ)+YY(IMJ-1)-YY(IJ-1))
      DNE(J)=DELTA(DXB,DYB,DXBP,DYBP)
   18 CONTINUE
C
```

```
C---- CALCULATE CELL VOLUMES
C
      DO 19 IJ=1,NINJ
      VOL(IJ)=ARE(IJ)
   19 CONTINUE
C
      IF(AKSI) THEN
      SIXR=1./6.
      DO 20 I=2,NIM
      II=IMNJ(I)
      DO 21 J=2,NJM
      IJ=II+J
      IMJ=IJ-NJ
      IMJM=IMJ-1
      IJM=IJ-1
      RIJ=YY(IJ)**2
      RIMJ=YY(IMJ)**2
      RIMJM=YY(IMJM)**2
      RIJM=YY(IJM)**2
      VOL(IJ)=SIXR*((XX(IJ)-XX(IMJ))*(RIJ+RIMJ+YY(IJ)*YY(IMJ))+
     &        (XX(IMJ)-XX(IMJM))*(RIMJ+RIMJM+YY(IMJ)*YY(IMJM))+
     &        (XX(IMJM)-XX(IJM))*(RIMJM+RIJM+YY(IMJM)*YY(IJM))+
     &        (XX(IJM)-XX(IJ))*(RIJM+RIJ+YY(IJM)*YY(IJ)))
   21 CONTINUE
   20 CONTINUE
      ENDIF
C
C---- INITIALIZE VARIABLES INITIALLY
C
      HAF=0.5
      QTR=0.25
      SMALL=1.E-30
      GREAT=1.E30
      DO 22 IJ=1,NINJ
      DEN(IJ)=DENSIT
      VIS(IJ)=VISCOS
      FMU(IJ)=1.0
      FLR1(IJ)=1.0
      FLR2(IJ)=1.0
      APU(IJ)=0.0
      APV(IJ)=0.0
      AE(IJ)=0.0
      AS(IJ)=0.0
      AN(IJ)=0.0
      AW(IJ)=0.0
      BE(IJ)=0.0
      BW(IJ)=0.0
      BN(IJ)=0.0
      BS(IJ)=0.0
      RES(IJ)=0.0
      R(IJ)=1.0
22    CONTINUE
      IF(AKSI) THEN
      DO 23 IJ=1,NINJ
      R(IJ)=YY(IJ)
23    CONTINUE
      ENDIF
C
      RETURN
      END
C
C---------------------------------------------------
      SUBROUTINE CALCE (PHI,IPHI)
C---------------------------------------------------
      INCLUDE 'kemod.h'
C
      DIMENSION UW(NY),VW(NY),WW(NY),PHI(NXNY),FXWW(NY),DW(NY)
C
      IF(IPHI.EQ.1) THEN
      URFPHI=1./URFK
      PRTINVP=1./PRTK
```



```
      ELSE
      URFPHI=1./URFE
      PRTINVP=1./PRTE
      ENDIF
C
      IJ=1
      PHINE=PHI(IJ)
      PHINW(IJ)=PHINE
C
      DO 30 J=2,NJM
      IJ=J
      IJM=IJ-1
      IPJ=IJ+NJ
      IJP=IJ+1
      FYN=FY(IJ)
      FYS=1.0-FYN
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      DXE=XX(IJ)-XX(IJM)
      DYE=YY(IJ)-YY(IJM)
      VIST=VIS(IJ)-VISCOS
      GAME=HAF*(VISCOS+VIST*PRTINVP)*(R(IJ)+R(IJM))
      DW(J)=GAME/AREE*(DXE**2+DYE**2)
      PHISE=PHINE
      PHINE=PHI(IJ+1)*FYN+PHI(IJ)*FYS
      PHINW(J)=PHINE
      UW(J)=U(IJ)
      VW(J)=V(IJ)
      WW(J)=W(IJ)
      SNSW(J)=0.
      FXWW(J)=1.0
      IF(JTBW(J).EQ.3.OR.JTBW(J).EQ.4) GO TO 30
      DXKS=QTR*(XX(IPJ)+XX(IPJ-1)-XX(IJ)-XX(IJM))
      DYKS=QTR*(YY(IPJ)+YY(IPJ-1)-YY(IJ)-YY(IJM))
      SNSW(J)=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
   30 CONTINUE
C
      DO 32 I=2,NIM
      J=1
      IJ=IMNJ(I)+J
      IMJ=IJ-NJ
      IJP=IJ+1
      FXE=FX(IJ)
      FXW=1.-FX(IJ)
      AREN=HAF*(ARE(IJ)+ARE(IJP))
      DXN=XX(IJ)-XX(IMJ)
      DYN=YY(IJ)-YY(IMJ)
      VIST=VIS(IJ)-VISCOS
      GAMN=HAF*(VISCOS+VIST*PRTINVP)*(R(IJ)+R(IMJ))
      DN=GAMN/AREN*(DXN**2+DYN**2)
      FYSS=1.0
      PHINE=PHI(IJ+NJ)*FXE+PHI(IJ)*FXW
      UN=U(IJ)
      VN=V(IJ)
      WN=W(IJ)
      SEWN=0.
      IF(ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 33
      DXET=QTR*(XX(IJP)+XX(IJP-NJ)-XX(IJ)-XX(IMJ))
      DYET=QTR*(YY(IJP)+YY(IJP-NJ)-YY(IJ)-YY(IMJ))
      SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
      SEWNBS=SEWN
   33 CONTINUE
      PHINW(J)=PHINE
C
C-----THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES
C
      DO 34 J=2,NJM
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      IJP=IJ+1
      IJM=IJ-1
```

```
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.-FYN
      DXE=XX(IJ)-XX(IJM)
      DYE=YY(IJ)-YY(IJM)
      DXN=XX(IJ)-XX(IMJ)
      DYN=YY(IJ)-YY(IMJ)
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      AREN=HAF*(ARE(IJ)+ARE(IJP))
      VISE=VIS(IJ)*FXW+VIS(IPJ)*FXE
      VISET=VISE-VISCOS
      GAME=HAF*(VISCOS+VISET*PRTINVP)*(R(IJ)+R(IJM))
      VISN=VIS(IJ)*FYS+VIS(IJP)*FYN
      VISNT=VISN-VISCOS
      GAMN=HAF*(VISCOS+VISNT*PRTINVP)*(R(IJ)+R(IMJ))
C
      DS=DN
      DE=GAME/AREE*(DXE**2+DYE**2)
      DN=GAMN/AREN*(DXN**2+DYN**2)
C
C LINEAR UPWIND DIFFERENCING
C
      AEE=MIN(F1(IJ),0.0)*FX(IPJ)*G
      AWW=-MAX(F1(IMJ),0.0)*(1.0-FXWW(J))*G
      AE1=-MIN(F1(IMJ),0.0)*FXE*G
      AW1=MAX(F1(IJ),0.0)*(1.0-FX(IMJ))*G
      ANN=MIN(F2(IJ),0.0)*FY(IJP)*G
      ASS=-MAX(F2(IJM),0.0)*(1.0-FYSS)*G
      AN1=-MIN(F2(IJM),0.0)*FYN*G
      AS1=MAX(F2(IJ),0.0)*(1.0-FY(IJM))*G
C
      AW(IJ)=DW(J)+MAX(F1(IMJ),0.0)-AWW
      AE(IJ)=DE-MIN(F1(IJ),0.0)-AEE
      AS(IJ)=DS+MAX(F2(IJM),0.0)-ASS
      AN(IJ)=DN-MIN(F2(IJ),0.0)-ANN
C
      DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))
      DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))
      DXET=QTR*(XX(IJP)-XX(IJM)+XX(IJP-NJ)-XX(IJM-NJ))
      DYET=QTR*(YY(IJP)-YY(IJM)+YY(IJP-NJ)-YY(IJM-NJ)+2.0*STAG)
C
      PHISE=PHINE
      PHINE=(PHI(IJ)*FYS+PHI(IJP)*FYN)*FXW+
     &      (PHI(IPJ)*FYS+PHI(IPJ+1)*FYN)*FXE
      SEWS=SEWN
      SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
      SNSE=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
      IF(I.EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.
      IF(J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.
      APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS
C
C  LINEAR UPWIND DIFFERENCING
C
      IMJ1=IMJ-NJ
      IMJJ=MAX(1,IMJ1)
      APV(IJ)=APV(IJ)+AEE*PHI(IPJ+NJ)+AWW*PHI(IMJJ)+ANN
     &       *PHI(IJP+1)+ASS*PHI(IJM-1)
     &       +AE1*PHI(IPJ)+AW1*PHI(IMJ)+AN1*PHI(IJP)+AS1*PHI(IJM)
      APU(IJ)=AEE+AWW+ANN+ASS+AE1+AW1+AN1+AS1
C
      VISS=VIS(IJ)-VISCOS+SMALL
C
      GO TO (37,38) IPHI
C
C-----TURBULENT KINETIC ENERGY SOURCE TERMS
C
   37 CONTINUE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
```



```
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))
      US=UN
      VS=VN
      WS=WN
      UN=U(IJP)*FYN+U(IJ)*FYS
      VN=V(IJP)*FYN+V(IJ)*FYS
      WN=W(IJP)*FYN+W(IJ)*FYS
      UE=U(IPJ)*FXE+U(IJ)*FXW
      VE=V(IPJ)*FXE+V(IJ)*FXW
      WE=W(IPJ)*FXE+W(IJ)*FXW
      DUEW=UE-UW(J)
      DUNS=UN-US
      DVEW=VE-VW(J)
      DVNS=VN-VS
      DWEW=WE-WW(J)
      DWNS=WN-WS
      GTERM=(2.*((DUEW*DYNS-DUNS*DYEW)**2+
     &          (DVNS*DXEW-DVEW*DXNS)**2)+(DUNS*DXEW-DUEW*DXNS+
     &           DVEW*DYNS-DVNS*DYEW)**2)/(ARE(IJ)**2)
       IF (AKSI)
     & GTERM=GTERM+((DYNS*DWEW-DYEW*DWNS)**2+
     &                (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &                /(ARE(IJ)**2)
     &              +2.*(V(IJ)/RP)**2

       DIVUV=(DUEW*DYNS-DUNS*DYEW+DVNS*DXEW-DVEW*DXNS)/ARE(IJ)
       GTERM=GTERM-2./3.*DIVUV**2
       GTERM=MAX(GTERM,0.)
C
C    GENERATION OF KINETIC ENERGY
C
       GEN(IJ)=(VIS(IJ)-VISCOS)*GTERM
       SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
C
       BP(IJ)=APU(IJ)+VOL(IJ)*CMU*DEN(IJ)*DEN(IJ)*TE(IJ)*FMU(IJ)/
     &        VISS
C
       IF(WOMG.NE.0.) THEN
       DUDY(IJ)=(DUNS*DXEW-DUEW*DXNS)/ARE(IJ)
       DVDX(IJ)=(DVEW*DYNS-DVNS*DYEW)/ARE(IJ)
       ENDIF
C
       GO TO 40
C
C-----DISSIP.OF TURB. KIN. ENERGY SOURCE TERMS
C
   38  CONTINUE
C
       SU(IJ)=APV(IJ)+FLR1(IJ)*C1*CMU*GEN(IJ)*DEN(IJ)*TE(IJ)*VOL(IJ)
     &       *FMU(IJ)/VISS
       BP(IJ)=APU(IJ)+FLR2(IJ)*FMU(IJ)*C2*CMU*DEN(IJ)**2*TE(IJ)*VOL(IJ)
     &        /VISS
C
       RI=0.
       IF(WOMG.NE.0.) THEN
       VOR=DUDY(IJ)-DVDX(IJ)
       RI=(0.4*VOR-0.8*WOMG)*WOMG*(TE(IJ)/(ED(IJ)+SMALL))**2
       END IF
       FLR2R=FLR2(IJ)+RI
               BP(IJ)=APU(IJ)+(FLR2R*C2*CMU*FMU(IJ)*DEN(IJ)*DEN(IJ)
     1                 *TE(IJ)/VISS
C
   40  CONTINUE
       UW(J)=UE
       VW(J)=VE
       WW(J)=WE
       SNSW(J)=SNSE
       PHINW(J)=PHINE
       FYSS=FY(IJM)
       FXWW(J)=FX(IMJ)
```

```
      DW(J)=DE
C
   34 CONTINUE
   32 CONTINUE
C
C-----PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS
C
      IDIR=IPHI
      CALL MODPHI
C
      IF(IPHI.EQ.2 .AND. LAY2 ) THEN
      DO 41 I=2,NIM
      IF(ITBS(I).NE.4) GO TO 42
      DO 43 J=2,J2LS
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
43    CONTINUE
42    IF(ITBN(I).NE.4) GO TO 41
      DO 44 J=J2LN,NJM
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
44    CONTINUE
41    CONTINUE
C
      DO 50 J=2,NJM
      IF(JTBW(J).NE.4) GO TO 51
      DO 52 I=2,I2LW
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
52    CONTINUE
51    IF(JTBE(J).NE.4) GO TO 50
      DO 53 I=I2LE,NIM
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
53    CONTINUE
50    CONTINUE
      ENDIF
C
      DO 60 I=2,NIM
      DO 60 J=2,NJM
      IJ=IMNJ(I)+J
      AP(IJ)=AW(IJ)+AE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)
      AP(IJ)=AP(IJ)*URFPHI
      SU(IJ)=SU(IJ)+(1.-URF(IPHI))*AP(IJ)*PHI(IJ)
   60 CONTINUE
C
C-----SOLVING F.D. EQUATIONS
C
      CALL SOLSIP(PHI,IPHI)
C
      RETURN
      END
C
C--------------------------------------------------------
      SUBROUTINE TWOLAY
C--------------------------------------------------------
      INCLUDE 'kemod.h'
C
C
C...ALONG THE SOUTH BOUNDARY
C
      DO 70 I=2,NIM
      IF(ITBS(I).NE.4) GO TO 70
      DISNS=GREAT
      DISNW=GREAT
      DISNE=GREAT
C
```



```
      DO 71 J=2,J2LS
      IJ=IMNJ(I)+J
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XPW=HAF*(XX(IJW)+XX(IMJW))
      YPW=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XPW
      DYBP=YBP-YPW
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK WEST BOUNDARY
      IF(JTBW(J).EQ.4) THEN
      IJW=J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNW=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C--CHECK EAST BOUNDARY
      IF(JTBE(J).EQ.4) THEN
      IJW=IMNJ(NIM)+J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNE=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C
      DISN=MIN(DISNS,DISNW,DISNE)
      RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS
      IF(LAY2) THEN
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      IF(WOMG.NE.0.) THEN
      AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WOMG)*WOMG*
     #            (TE(IJ)/(ED(IJ)+SMALL))**2
      AME=ABS(AME)
      ALMU=ALMU*AME**1.5
      ALED=ALED*AME**0.5
      END IF
      ED(IJ)=SQRT(TE(IJ))**3/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU
          ELSE
      RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))
      FMU(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RK))**2
      FLR1(IJ)=1.0+(0.05/FMU(IJ))**3
      FLR2(IJ)=1.0-EXP(-RT*RT)
            ENDIF
71    CONTINUE
70    CONTINUE
C
C...ALONG THE NORTH BOUNDARY
C
      DO 72 I=2,NIM
      IF(ITBN(I).NE.4) GO TO 72
      DISNN=GREAT
      DISNW=GREAT
```

```
      DISNE=GREAT
      DO 73 J=J2LN,NJM
      IJ=IMNJ(I)+J
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK WEST BOUNDARY
      IF(JTBW(J).EQ.4) THEN
      IJW=J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNW=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C--CHECK EAST BOUNDARY
      IF(JTBE(J).EQ.4) THEN
      IJW=IMNJ(NIM)+J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNE=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C
      DISN=MIN(DISNN,DISNW,DISNE)
      RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS
      IF(LAY2) THEN
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      IF(WOMG.NE.0.) THEN
      AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WOMG)*WOMG*
     #                      (TE(IJ)/(ED(IJ)+SMALL))**2
      AME=ABS(AME)
      ALMU=ALMU*AME**1.5
      ALED=ALED*AME**0.5
      END IF
      ED(IJ)=SQRT(TE(IJ))**3/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU
        ELSE
      RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))
      FMU(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RK))**2
      FLR1(IJ)=1.0+(0.05/FMU(IJ))**3
      FLR2(IJ)=1.0-EXP(-RT*RT)
           ENDIF
73    CONTINUE
72    CONTINUE
C
C
C...ALONG THE WEST BOUNDARY
C
      DO 75 J=2,NJM
      IF(JTBW(J).NE.4) GO TO 75
```



```
      DISNW=GREAT
      DISNN=GREAT
      DISNS=GREAT
      DO 76 I=2,I2LW
      IJ=IMNJ(I)+J
      IJW=J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNW=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK NORTH BOUNDARY
      IF(ITBN(I).EQ.4) THEN
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C--CHECK SOUTH BOUNDARY
      IF(ITBS(I).EQ.4) THEN
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XPW=HAF*(XX(IJW)+XX(IMJW))
      YPW=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XPW
      DYBP=YBP-YPW
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C
      DISN=MIN(DISNW,DISNS,DISNN)
      RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS
      IF(LAY2) THEN
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      IF(WOMG.NE.0.) THEN
      AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WOMG)*WOMG*
     #                      (TE(IJ)/(ED(IJ)+SMALL))**2
      AME=ABS(AME)
      ALMU=ALMU*AME**1.5
      ALED=ALED*AME**0.5
      END IF
      ED(IJ)=SQRT(TE(IJ))**3/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU
        ELSE
      RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))
      FMU(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RK))**2
      FLR1(IJ)=1.0+(0.05/FMU(IJ))**3
      FLR2(IJ)=1.0-EXP(-RT*RT)
           ENDIF
76    CONTINUE
75    CONTINUE
C
C...ALONG THE EAST BOUNDARY
C
      DO 78 J=2,NJM
```

```
      IF(JTBE(J).NE.4) GO TO 78
      DISNE=GREAT
      DISNN=GREAT
      DISNS=GREAT
      DO 79 I=I2LE,NIM
      IJ=IMNJ(I)+J
      IJW=IMNJ(NIM)+J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNE=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK NORTH BOUNDARY
      IF(ITBN(I).EQ.4) THEN
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C--CHECK SOUTH BOUNDARY
      IF(ITBS(I).EQ.4) THEN
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XPW=HAF*(XX(IJW)+XX(IMJW))
      YPW=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XPW
      DYBP=YBP-YPW
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C
      DISN=MIN(DISNE,DISNS,DISNN)
      RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS
      IF(LAY2) THEN
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      IF(WOMG.NE.0.) THEN
      AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WOMG)*WOMG*
     #                          (TE(IJ)/(ED(IJ)+SMALL))**2
      AME=ABS(AME)
      ALMU=ALMU*AME**1.5
      ALED=ALED*AME**0.5
      END IF
      ED(IJ)=SQRT(TE(IJ))**3/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU
        ELSE
      RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))
      FMU(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RK))**2
      FLR1(IJ)=1.0+(0.05/FMU(IJ))**3
      FLR2(IJ)=1.0-EXP(-RT*RT)
          ENDIF
 79   CONTINUE
 78   CONTINUE
C
      RETURN
      END
```



```
C
      FUNCTION DELTA(DXB,DYB,DXBP,DYBP)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      DELP=DXB*DXBP+DYB*DYBP
      DELTA=SQRT(DXBP**2+DYBP**2-DELP**2)
      RETURN
      END
C
C------------------------------------------------------------
      SUBROUTINE SOLSIP(PHI,IPHI)
C------------------------------------------------------------
      INCLUDE 'kemod.h'
C
      DIMENSION PHI(NXNY)
      ID=IPHI
      L=0
      DO 5 I=2,NIM
      DO 5 J=2,NJM
      IJ=IMNJ(I)+J
      API=1.0/AP(IJ)
      AP(IJ)=1.0
      AE(IJ)=AE(IJ)*API
      AW(IJ)=AW(IJ)*API
      AN(IJ)=AN(IJ)*API
      AS(IJ)=AS(IJ)*API
      SU(IJ)=SU(IJ)*API
5     CONTINUE
C
      DO 10 I=2,NIM
      DO 11 J=2,NJM
      IJ=IMNJ(I)+J
      IJM=IJ-1
      IMJ=IJ-NJ
      BW(IJ)=-AW(IJ)/(1.+ALFA*BN(IMJ))
      BS(IJ)=-AS(IJ)/(1.+ALFA*BE(IJM))
      POM1=ALFA*BW(IJ)*BN(IMJ)
      POM2=ALFA*BS(IJ)*BE(IJM)
      BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IMJ)-BS(IJ)*BN(IJM)
      BN(IJ)=(-AN(IJ)-POM1)/(BP(IJ)+SMALL)
      BE(IJ)=(-AE(IJ)-POM2)/(BP(IJ)+SMALL)
   11 CONTINUE
   10 CONTINUE
      NSTP=NSWP(ID)
  100 CONTINUE
      L=L+1
      RESORP=0.
      DO 20 I=2,NIM
      DO 21 J=2,NJM
      IJ=IMNJ(I)+J
      RES(IJ)=AN(IJ)*PHI(IJ+1)+AS(IJ)*PHI(IJ-1)+AE(IJ)*PHI(IJ+NJ)+
     &          AW(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)
      RESORP=RESORP+ABS(RES(IJ))
      RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/
     &          (BP(IJ)+SMALL)
   21 CONTINUE
   20 CONTINUE
      IF(L.EQ.1) RESOR(ID)=RESORP
      RSM=SOR(ID)*RESOR(ID)
      DO 30 I=2,NIM
      II=NIM+2-I
      DO 31 J=2,NJM
      JJ=NJM+2-J
      IJ=IMNJ(II)+JJ
      RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)
      PHI(IJ)=PHI(IJ)+RES(IJ)
   31 CONTINUE
   30 CONTINUE
      IF(TEST) PRINT 1, L,RESORP
      IF(RESORP.GT.RSM.AND.L.LT.NSTP) GO TO 100
```

```
      IF(RESORP.GT.RSM.AND.L.GE.NSTP) WRITE(6,2)
    1 FORMAT(10X,I5,' SWEEP, RESOR =',E12.4)
    2 FORMAT(//,10X,' SIPSOL DID NOT CONVERGE  ',//)
      RETURN
      END
C
C----------------------------------------------------------
      SUBROUTINE USERM
C----------------------------------------------------------
      INCLUDE 'kemod.h'
C
C
      DATA CAPPA/0.4197/
C
C-----------------
      ENTRY MODVIS
C-----------------
      DO 80 I=1,NI
      DO 80 J=1,NJ
      IJ=IMNJ(I)+J
      VISOLD=VIS(IJ)
      VIS(IJ)=VISCOS
      IF(ED(IJ).GT.SMALL)
     &  VIS(IJ)=FMU(IJ)*DEN(IJ)*TE(IJ)**2*CMU/ED(IJ)+VISCOS
      VIS(IJ)=URFVIS*VIS(IJ)+(1.-URFVIS)*VISOLD
   80 CONTINUE
C
      IF(LAY2) THEN
      DO 81 I=2,NIM
      IF(ITBS(I).NE.4) GO TO 82
      DO 83 J=2,J2LS
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
83    CONTINUE
82    IF(ITBN(I).NE.4) GO TO 81
      DO 84 J=J2LN,NJM
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
84    CONTINUE
81    CONTINUE
      DO 85 J=2,NJM
      IF(JTBE(J).NE.4) GO TO 86
      DO 87 I=I2LE,NIM
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
87    CONTINUE
86    IF(JTBW(J).NE.4) GO TO 85
      DO 88 I=2,I2LW
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
88    CONTINUE
85    CONTINUE
C
      ENDIF
C
      RETURN

C-----------------------
      ENTRY MODPHI
C-----------------------
      GO TO (800,900) IDIR
C
C-----BOUNDARY CONDITIONS FOR KINETIC TURBULENT ENERGY
C
  800 CONTINUE
C-----SOUTH BOUNDARY
      DO 810 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (811,812,813,814) ITBS(I)
  811 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)
```



```
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 815
  812 TE(IJ-1)=TE(IJ)
      GO TO 815
  813 CONTINUE
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
      TE(IJ)=TE(IJ+1)-DEL*FAC
      IJ=IJ+1
      GO TO 815
  814 CONTINUE
      IF(.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=GENTS(I)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      WN=W(IJP)*FYN+W(IJ)*FYS
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &                (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &                /(ARE(IJ)**2)
     &             +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  815 CONTINUE
      AS(IJ)=0.0
  810 CONTINUE
C
C-----NORTH BOUNDARY
      DO 820 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (821,822,823,824) ITBN(I)
  821 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 825
  822 TE(IJ+1)=TE(IJ)
      GO TO 825
  823 CONTINUE
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
```

```
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
      TE(IJ)=TE(IJ-1)-DEL*FAC
      IJ=IJ-1
      GO TO 825
  824 CONTINUE
      IF(.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=GENTN(I)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      WN=W(IJP)*FYN+W(IJ)*FYS
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &              (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &              /(ARE(IJ)**2)
     &            +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  825 CONTINUE
      AN(IJ)=0.0
  820 CONTINUE
C-----WEST BOUNDARY
      DO 830 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (831,832,833,834) JTBW(J)
  831 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*TE(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 835
  832 TE(IJ-NJ)=TE(IJ)
      GO TO 835
  833 CONTINUE
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2
      TE(IJ)=TE(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GO TO 835
  834 CONTINUE
      IF(.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=GENTW(J)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
```



```
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      WN=W(IJP)*FYN+W(IJ)*FYS
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &              (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &              /(ARE(IJ)**2)
     &            +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  835 CONTINUE
      AW(IJ)=0.0
  830 CONTINUE
C-----EAST BOUNDARY
      DO 840 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (841,842,843,844) JTBE(J)
  841 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*TE(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 845
  842 TE(IJ+NJ)=TE(IJ)
      GO TO 845
  843 CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2
      TE(IJ)=TE(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 845
  844 CONTINUE
      IF(.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=GENTEE(J)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      WN=W(IJP)*FYN+W(IJ)*FYS
```

```
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &                 (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &                 /(ARE(IJ)**2)
     &              +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  845 CONTINUE
      AE(IJ)=0.0
  840 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR DISSIPATION OF KIN. TURB. ENERGY
C
  900 CONTINUE
C
      CMU25=SQRT(SQRT(CMU))
      CMU75=CMU25**3
C
C-----SOUTH BOUNDARY
      DO 910 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (911,912,913,914) ITBS(I)
  911 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 915
  912 ED(IJ-NJ)=ED(IJ)
      GO TO 915
  913 CONTINUE
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ+1)-DEL*FAC
      IJ=IJ+1
      GO TO 915
  914 CONTINUE
      IF(LRE) THEN
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ+1)-DEL*FAC
      IJ=IJ+1
      ELSE
C
      TE(IJ)=ABS(TE(IJ))
```



```
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNS(I))*GREAT
      BP(IJ)=GREAT
      ENDIF
  915 CONTINUE
      AS(IJ)=0.0
  910 CONTINUE
C-----NORTH BOUNDARY
      DO 920 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (921,922,923,924) ITBN(I)
  921 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 925
  922 ED(IJ+1)=ED(IJ)
      GO TO 925
  923 CONTINUE
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ-1)-DEL*FAC
      IJ=IJ-1
      GO TO 925
  924 CONTINUE
      IF(LRE) THEN
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ-1)-DEL*FAC
      IJ=IJ-1
      ELSE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNN(I))*GREAT
      BP(IJ)=GREAT
      ENDIF
  925 CONTINUE
      AN(IJ)=0.0
  920 CONTINUE
C-----WEST BOUNDARY
      DO 930 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (931,932,933,934) JTBW(J)
  931 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 935
  932 ED(IJ-NJ)=ED(IJ)
      GO TO 935
  933 CONTINUE
      IJ=J
      IJP=IJ+1
```

```
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GO TO 935
  934 CONTINUE
      IF(LRE) THEN
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      ELSE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNW(J))*GREAT
      BP(IJ)=GREAT
      ENDIF
  935 CONTINUE
      AW(IJ)=0.0
  930 CONTINUE
C-----EAST BOUNDARY
      DO 940 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (941,942,943,944) JTBE(J)
  941 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 945
  942 ED(IJ+NJ)=ED(IJ)
      GO TO 945
  943 CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 945
  944 CONTINUE
      IF(LRE) THEN
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
```



```
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      ELSE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNE(J))*GREAT
      BP(IJ)=GREAT
      ENDIF
  945 CONTINUE
      AE(IJ)=0.0
  940 CONTINUE
C
      END
C
C--- SUBROUTINE 'MODIFY' TO MODIFY SHEAR STRESSES IN
C--- NEAR WALL GRID POINT
C--------------------------------------------------------
      SUBROUTINE MODIFY (SUR,BPR)
C--------------------------------------------------------
      INCLUDE 'kemod.h'
      DIMENSION SUR(NXNY),BPR(NXNY)
C
      DATA CMU25,CAPPA,ELOG/0.5477,0.4197,9.0/
C
      DO 10 I=1,NI
      DO 10 J=1,NJ
      IJ=IMNJ(I)+J
      SU(IJ)=SUR(IJ)
      BP(IJ)=BPR(IJ)
   10 CONTINUE
C
C--- CHECK WALL SOUTH BOUNDARY
C
      DO 600 I=2,NIM
      IJ=IMNJ(I)+2
      IF(ITBS(I).EQ.4) THEN
      LB=IJ
      LW=IJ-1
      TEPR=SQRT(TE(IJ))
      DELN=DNS(I)
      DXB=XX(IJ-1)-XX(IJ-NJ-1)
      DYB=YY(IJ-1)-YY(IJ-NJ-1)
      RB=HAF*(R(IJ-1)+R(IJ-NJ-1))
      DENS=DEN(IJ)
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
      SUVS(I)=SVU
      SPVS(I)=SVP
      SUWS(I)=SWU
      SPWS(I)=SWP
      GENTS(I)=GENTE
      IF(LRE) THEN
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
```

```
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ+1)-DEL*FAC
      IJ=IJ+1
      ENDIF
      AS(IJ)=0.0
      ENDIF
C
C-- CHECK NORTH WALL-BOUNDARY
C
      IJ=IMNJ(I)+NJM
      IF(ITBN(I).EQ.4) THEN
      LB=IJ
      LW=IJ+1
      TEPR=SQRT(TE(IJ))
      DELN=DNN(I)
      DXB=XX(IJ)-XX(IJ-NJ)
      DYB=YY(IJ)-YY(IJ-NJ)
      RB=HAF*(R(IJ)+R(IJ-NJ))
      DENS=DEN(IJ)
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
      SUVN(I)=SVU
      SPVN(I)=SVP
      SUWN(I)=SWU
      SPWN(I)=SWP
      GENTN(I)=GENTE
      IF(LRE) THEN
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ-1)-DEL*FAC
      IJ=IJ-1
      ENDIF
      AN(IJ)=0.0
      ENDIF
C
600   CONTINUE
C
C-- CHECK WALL WEST-BOUNDARY
C
      DO 620 J=2,NJM
      IJ=IMNJ(2)+J
      IF(JTBW(J).EQ.4) THEN
      LB=IJ
      LW=IJ-NJ
      TEPR=SQRT(TE(IJ))
      DELN=DNW(J)
      DXB=XX(IJ-NJ)-XX(IJ-NJ-1)
      DYB=YY(IJ-NJ)-YY(IJ-NJ-1)
      RB=HAF*(R(IJ-NJ)+R(IJ-NJ-1))
      DENS=DEN(IJ)
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
```



```
      SUVW(J)=SVU
      SPVW(J)=SVP
      SPWW(J)=SWP
      SUWW(J)=SWU
      GENTW(J)=GENTE
      IF(LRE) THEN
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      ENDIF
      AW(IJ)=0.0
      ENDIF
C
C-- CHECK WALL EAST-BOUNDARY
C
      IJ=IMNJ(NIM)+J
      IF(JTBE(J).EQ.4) THEN
      LB=IJ
      LW=IJ+NJ
      TEPR=SQRT(TE(IJ))
      DELN=DNE(J)
      DXB=XX(IJ)-XX(IJ-1)
      DYB=YY(IJ)-YY(IJ-1)
      RB=HAF*(R(IJ)+R(IJ-1))
      DENS=DEN(IJ)
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
      SUVE(J)=SVU
      SPVE(J)=SVP
      SPWE(J)=SWP
      SUWE(J)=SWU
      GENTEE(J)=GENTE
      IF(LRE) THEN
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      ENDIF
      AE(IJ)=0.0
      ENDIF
C
620   CONTINUE
C
      RETURN
      END
C
```

```
C
C--- SUBROUTINE 'WALLFN' TO SET WALL FUNCTIONS
C
      SUBROUTINE WALLFN (LB,LW,VISC,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
C
      INCLUDE 'kemod.h'
C
      UP=U(LB)
      VP=V(LB)
      WP=W(LB)
      UWALL=U(LW)
      VWALL=V(LW)
      WWALL=W(LW)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      CONST=DENS*CMU25*TEPR
      YPLS=DELN*CONST/VISC
      TCOEF=VISC/DELN
      IF(LAY2.OR.LRE) GOTO 10
      IF (YPLS.LE.11.63) GO TO 10
      UPLUS=LOG(ELOG*YPLS)/CAPPA
      TCOEF=CONST/UPLUS
   10 CONTINUE
      VPINT=UP*DXB+VP*DYB
      VPX=VPINT*DXB
      VPY=VPINT*DYB
      TAU=-TCOEF*(VPX*DXB+VPY*DYB)
      ARW=ARW*RB
      SUP=TCOEF*ARW*DXB**2
      SVP=TCOEF*ARW*DYB**2
      SWP=TCOEF*ARW
      CON=TCOEF*ARW*DXB*DYB
      SUU=-CON*VP+UWALL*TCOEF*ARW*DXB**2
      SVU=-CON*UP+VWALL*TCOEF*ARW*DYB**2
      SWU= TCOEF*ARW*WWALL*RB
C
C    FOR MOVING WALL
C
      VPINT=VPINT+WP
      VPINT=ABS(VPINT-SQRT(UWALL*UWALL+VWALL*VWALL+WWALL*WWALL))
C
      GENTE=TCOEF*CONST*ABS(VPINT)/(CAPPA*DENS*DELN)
C
      RETURN
      END
C
C-----------------------------------------------------------------
C    kemod.h
C
C    COMMON BLOCK
      PARAMETER (NX=75)
      PARAMETER (NY=50)
      PARAMETER (NXNY=NX*NY)
      COMMON/KEA1/URFK,URFE,URFVIS,NI,NJ,NIM,NJM,NINJ,IMNJ(NX),
     &   IDIR,ITER,ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY),
     &   TEST,AKSI,G
      COMMON/KEA2/XX(NXNY),YY(NXNY),FX(NXNY),FY(NXNY),ARE(NXNY),
     &   VOL(NXNY),DNS(NX),DNN(NX),DNW(NY),DNE(NY),R(NXNY)
      COMMON/KEA3/AE(NXNY),AW(NXNY),AN(NXNY),AS(NXNY),AP(NXNY),
     &   SU(NXNY)
      COMMON/KEA4/APU(NXNY),APV(NXNY),BE(NXNY),BW(NXNY),BN(NXNY),
     &   BS(NXNY),BP(NXNY),RES(NXNY),F1(NXNY),F2(NXNY)
      COMMON/KEA5/U(NXNY),V(NXNY),W(NXNY),P(NXNY),GEN(NXNY),
     &   TE(NXNY),ED(NXNY),DEN(NXNY),VIS(NXNY),RW(NXNY)
      COMMON/KEA6/RESOR(2),PRTK,PRTE,URF(2),SORMAX,
     &   VISCOS,C1,C2,CMU,ALFA,GREAT,SMALL,SNSW(NY),
     &   UNW(NY),VNW(NY),PHINW(NY),HAF,QTR,DENSIT
      COMMON/KEAA/NSWP(2),SOR(2)
      COMMON/TURB/ FLR1(NXNY),FLR2(NXNY),VIS2(NXNY),FMU(NXNY),
```



```
     &   J2LS,J2LN,I2LE,I2LW,LRE,LAY2
      COMMON/KEUVMOD/ SUVS(NX),SPVS(NX),SUVN(NX),SPVN(NX),
     &                SUVW(NY),SPVW(NY),SUVE(NY),SPVE(NY),
     &                SPWN(NX),SPWS(NX),SPWE(NY),SPWW(NY),
     &                SUWN(NX),SUWS(NX),SUWE(NY),SUWW(NY)
      COMMON /KEGENER/ GENTS(NX),GENTN(NX),GENTW(NY),GENTEE(NY)
      COMMON /ROT/     WOMG,DUDY(NXNY),DVDX(NXNY)
      LOGICAL TEST,AKSI,LRE,LAY2
```

CHAPTER 3

2D/Axisymmetric Multi-Time-Scale k-ε Turbulence Model

Table of Contents

In this section a description of the multi-time-scale k-ε turbulence model that is coded as a self contained computer program to compute turbulent flow quantities in two-dimensional or axisymmetric geometry is given. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in Appendix B. The module has been tested as a separate self-contained unit using the REACT code [1] and was independently tested at the University of Alabama at Huntsville (UAH) using own code (MAST).

3.1 Introduction

Turbulent flows comprise fluctuating motions with a spectrum of sizes and time scales and different turbulent interactions are associated with different parts of the spectrum. In the single-time-scale turbulence models such as the k-ε turbulence model it is assumed that a single time scale (proportional to k/ε) can be used to describe the turbulent flow. In many complex flows turbulence is generally in spectral inequilibrium and a single time scale description is a simplification.

Figure 1, shows a sketch of a typical energy spectrum in a turbulent flow at high Reynolds number in a simplified split spectrum method. Two regions can be identified, the production range (at wave number $\kappa < \kappa_I$) where the kinetic energy (k_p) leaves this region at a rate (ε_p) and a high wave number or dissipation region ($\kappa > \kappa_I$) with kinetic energy (k_t) and energy dissipation rate (ε_t). Hanjalic et al. [2] developed a simple multiple-time-scale turbulence model based on a rational extension of the single scale equation ideas. In their model, a fixed ratio of the turbulent kinetic energy of large eddies (k_p) to that of the fine scale eddies (k_t) is used to partition the spectrum. Kim and Chen [3] improved on the simplified split spectrum by dynamically determining the location of the partition (i.e k_p/k_t) as part of the solution and is dependent on the turbulence intensity, production rate, energy transfer and dissipation rate. The variable partitioning method causes the effective eddy viscosity to decrease when production is high and to increase when production vanishes -a behavior consistent with experimental observations.

3.2 Theory and Model Equations

The multi-time-scale turbulence module is based on the variable partitioning of the turbulent energy spectrum proposed by Kim and Chen [3]. In this model the turbulent kinetic energy spectrum is divided into two sets of wave number regions giving two evolution equations for each region.

These equations represent the kinetic energy (k_p) and the energy transfer rate (ε_p) in the production range of the spectrum and the kinetic energy (k_t) and the energy dissipation rate (ε_t) in the dissipation range of the spectrum. This model allows the partition to move toward the high wave number region when production is high and toward the low wave number region when production vanishes.

The equations which describe the multi-time-scale turbulence model used are given below. The turbulent kinetic energy and the energy transfer rate equations for the energy containing large eddies are given as;

$$\rho \frac{Dk_p}{Dt} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_{k_p}}\right) \frac{\partial k_p}{\partial x_i} \right] + G - \rho\, \varepsilon_p \tag{1}$$

$$\rho \frac{D\varepsilon_p}{Dt} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_{\varepsilon_p}}\right) \frac{\partial \varepsilon_p}{\partial x_i} \right] + \frac{1}{\rho} C_{P1} \frac{G^2}{k_p} + C_{P2} \frac{G\varepsilon_p}{k_p} - \rho C_{P3} \frac{\varepsilon_p^2}{k_p} \tag{2}$$

where G is the turbulence production rate, given as

$$G = \mu_e \left\{ 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2 + \left(\frac{v}{r}\right)^2 \right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial r} - \frac{w}{r}\right)^2 \right\}$$

where μ is the viscosity

μ_t is the turbulent viscosity

k_p is the turbulent kinetic energy in the production range

ε_p is the energy transfer rate

σ_{k_p} and σ_{ε_p} are constants

C_{p1}, C_{p2} and C_{p3} are turbulent model constants

The turbulent kinetic energy and the dissipation rate equations for the high wave number small scale eddies region are given as;

$$\rho \frac{Dk_t}{Dt} = \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_t}{\sigma_{k_t}}\right) \frac{\partial k_t}{\partial x_i} \right] + \rho\, \varepsilon_p - \rho\, \varepsilon_t \tag{3}$$

$$\rho \frac{D\varepsilon_t}{Dt} = \frac{\partial}{\partial x_i} [(\mu + \frac{\mu_t}{\sigma_{\varepsilon_t}}) \frac{\partial \varepsilon_t}{\partial x_i}] + \rho\, C_{t1} \frac{\varepsilon_p^2}{k_t} + \rho\, C_{t2} \frac{\varepsilon_t\, \varepsilon_p}{k_t} - \rho\, C_{t3} \frac{\varepsilon_t^2}{k_t} \qquad (4)$$

where k_t is the turbulent kinetic energy in the dissipation range

ε_t is the energy dissipation rate

σ_{k_t} and σ_{ε_t} are constants

C_{t1}, C_{t2} and C_{t3} are turbulent model constants

The terms $\frac{1}{\rho} C_{P1} \frac{G^2}{k_p}$ and $\rho\, C_{t1} \frac{\varepsilon_p^2}{k_t}$ represent variable energy transfer functions. The first term increases the energy transfer rate when production is high and the second term increases the dissipation rate when the energy transfer rate is high. The turbulent viscosity is given as

$$\mu_t = \rho\, C_{\mu f} \frac{k^2}{\varepsilon_p} = \rho\, C_\mu \frac{k^2}{\varepsilon_t}$$

where $k = k_p + k_t$ is the total turbulent kinetic energy and $C_{\mu f}$ is a constant.

The model constants used are similar to those used by Kim and Chen [3]

$\sigma_{k_p} = 0.75,\ \sigma_{\varepsilon_p} = 1.15,\ \sigma_{k_t} = 0.75,\ \sigma_{\varepsilon_t} = 1.15$
$C_{P1} = 0.21,\ C_{P2} = 1.24,\ C_{P3} = 1.84,\ C_{t1} = 0.29$
$C_{t2} = 1.28,\ C_{t3} = 1.66$ and $C_{\mu f} = 0.09$

For turbulent flow analysis, equations (1)-(4) are solved by the module that is interfaced with a Reynolds averaged flow solver to compute the turbulent flow field. For an incompressible, steady and axisymmetric turbulent flow, a generalized equation that expresses the transport of turbulent flow can be written as;

$$\frac{\partial (\rho u \Phi)}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (\rho\, v\, r\, \Phi) = \frac{\partial}{\partial x} (\Gamma_{\Phi_x} \frac{\partial \Phi}{\partial x}) + \frac{1}{r} \frac{\partial}{\partial x} (r\, \Gamma_{\Phi_r} \frac{\partial \Phi}{\partial r}) + S_\Phi \qquad (5)$$

where Φ is the dependent variable, which stands for $\Phi = u$, v , w for the axial, radial and tangential velocities respectively. ρ is the fluid density, $\Gamma\Phi_x$ and $\Gamma\Phi_r$ are exchange coefficients in x and r -directions, respectively, and S_Φ is the source term for the variable Φ.

The source terms for the dependent variable are:

- Axial direction, $\Phi = u$, $\Gamma\Phi_x = 2\mu_e$, $\Gamma\Phi_r = \mu_e$ and

$$S_u = -\frac{\partial P}{\partial x} + \frac{1}{r}\frac{\partial}{\partial r}(\mu_e r \frac{\partial v}{\partial x})$$

where μ_e is the eddy viscosity and P is the pressure

- Radial direction, $\Phi = v$, $\Gamma\Phi_x = \mu_e$, $\Gamma\Phi_r = 2\mu_e$ and

$$S_v = -\frac{\partial}{\partial x}\left(\mu_e \frac{\partial u}{\partial r}\right) - 2\mu_e \frac{v}{r^2} + \frac{\rho w^2}{r} - \frac{\partial P}{\partial r}$$

- Tangential direction, $\Phi = w$, $\Gamma\Phi_x = \mu_e$, $\Gamma\Phi_r = \mu_e$ and

$$S_w = -\frac{\rho v w}{r} - \frac{w}{r^2}\frac{\partial}{\partial r}(r\,\mu_e)$$

Equations (1)-(4) can also be written in a similar form as equation (5) where Φ stands for;

- Turbulent kinetic energy in the production range of the energy spectrum

  $\Phi = k_p$, $\Gamma\Phi_x = \mu + \dfrac{\mu_t}{\sigma_{kp}} = \Gamma\Phi_r$ and

  $s_{kp} = G - \rho\,\varepsilon_p$

- Energy transfer rate in the production range of the energy spectrum

  $\Phi = \varepsilon_p$, $\Gamma\Phi_x = \mu + \dfrac{\mu_t}{\sigma_{\varepsilon p}} = \Gamma\Phi_r$ *and*

$$s_{\varepsilon p} = \frac{1}{\rho} C_{P1} \frac{G^2}{k_p} + C_{P2} \frac{G\varepsilon_p}{k_p} - \rho\, C_{P3} \frac{\varepsilon_p^2}{k_p}$$

- Turbulent kinetic energy in the dissipation range of the energy spectrum

$$\Phi = k_t, \ \Gamma\Phi_x = \mu + \frac{\mu_t}{\sigma_{k_t}} = \Gamma\Phi_r \ \ and$$

$$s_{k_t} = \rho\,\varepsilon_p \ - \ \rho\,\varepsilon_t$$

- Energy dissipation rate in the dissipation range of the energy spectrum

$$\Phi = \varepsilon_t, \ \Gamma\Phi_x = \mu + \frac{\mu_t}{\sigma_{\varepsilon_t}} = \Gamma\Phi_r \ \ and$$

$$s_{\varepsilon_t} = \rho\, C_{t1} \frac{\varepsilon_p^2}{k_t} + \rho\, C_{t2} \frac{\varepsilon_t \varepsilon_p}{k_t} - \rho\, C_{t3} \frac{\varepsilon_t^2}{k_t}$$

Near a wall, the wall function boundary conditions used are similar to that of Kim and Chen [3].

A two layer model for the multi-time-scale k-ε turbulence model similar to that of Chen and Patel [4] for the single-time-scale k-ε turbulence model is included in the present release.

3.3 Model Evaluation

The multi-time-scale k-ε module was evaluated by comparisons with experimental studies. One of the test problems considered was the backward facing step of Driver and Seegmiller [5] where the multi scale k-ε model predicted a recirculation length of 6.14 step heights (H) downstream of the step which is closer to the experimental value (6.10 H) than the standard k-ε model (5.35H).

The majority of the tests were conducted using Roback and Johnson's experimental data [6] for swirling confined double concentric jets. Preliminary analysis indicated some sensitivity to the ratio k_p / k_t at the inlet boundary, however, a value of 3 was found reasonable in the present analysis. Figures 2 and 3 show the streamline patterns for wall functions and two-layer near wall treatments respectively. The upper (a) and lower (b) parts correspond to the single-scale k-ε and the multi-scale k-ε models respectively. It can be seen from these contours that there are two recirculation zones in the chamber, one is near the expansion corner and another located in the central region and accurate predictions of this central region is very important in combusting swirling flows. Figures 4a and 4b, show the axial velocity along the centerline. In terms of strength and size of the central recirculation zone, the multi-scale k-ε model yields better agreement than the single-scale k-ε model. In the central recirculation region the k-ε model tends to connect the energy transfer rate to the local mean strain rate too strongly while the multi-scale model suppresses this tendency.

Figures 5 and 6 show the radial profiles of the mean axial velocity at different axial locations downstream of the inlet using the wall function and the two-layer near-wall treatments respectively. Similarly, figures 7 and 8 show the corresponding profiles for the tangential velocity, and figures 9 and 10 show the radial profiles of the axial normal turbulent intensity $(\overline{uu})^{1/2}$ using both the wall function and the two-layer near-wall treatments. In general, the numerical results indicate that the multi-scale model gives better agreement than the standard k-ε model.

REFERENCES

1. Chon, J., Hadid, A. and Hamakiotes C. "REACT-2D Version 2.6 User's Manual" CFD Technology Center, Rocketdyne Division/Rockwell International, 1989.

2. Hanjalic, K., Lauder, B. and Schiestel R. "Multiple-Time-Scale Concepts in Turbulent Transport Modeling" in Turbulent Shear Flows, Eds. Bradbury L. J. et al., vol. 2, Springer-Verlag, N.Y., pp. 36-49, 1980.

3. Kim, S.-W and Chen, C.-P "A Multiple-Time-Scale Turbulence Model Based on Variable Partitioning of the Turbulent Kinetic Energy Spectrum", Num. Heat Transfer, pt. B, vol. 16, no. 2, pp. 193-211, 1989.

4. Chen, H. C. and Patel, V. C. "Near-Wall Turbulence Models for Complex Flows Including Separation", AIAA Journal, vol. 26, no. 6, pp. 641-648, 1988.

5. Driver, D. and Seegmiller, H. "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow" AIAA J., vol. 23, pp. 163-171. 1985.

6. Roback, R. and Johnson, B "Mass and Momentum Turbulent Transport Experiment with Confined Swirling Co-axial Jets" NASA CR-168252, 1983.

7. Stone, H. "Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations", SIAM J. Num. Anal., vol. 5, pp 530-, 1968.

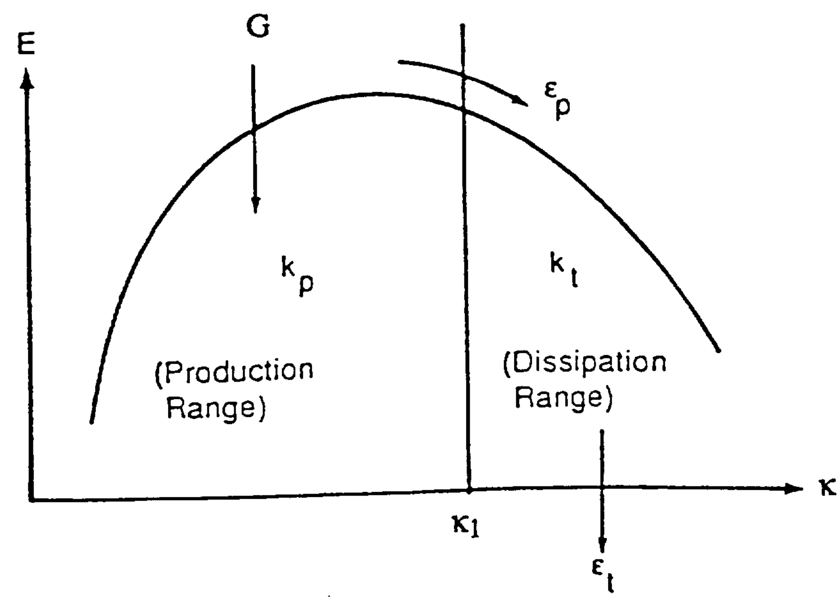


Figure 1. Description and nomenclature of the multiple-time-scale turbulence model;

$$k_p = \int_{\kappa=0}^{\kappa=\kappa_1} E \, d\kappa \ , \quad k_t = \int_{\kappa=\kappa_1}^{\kappa=\infty} E \, d\kappa$$

κ_1 = Partition wave number, E = Energy spectral density

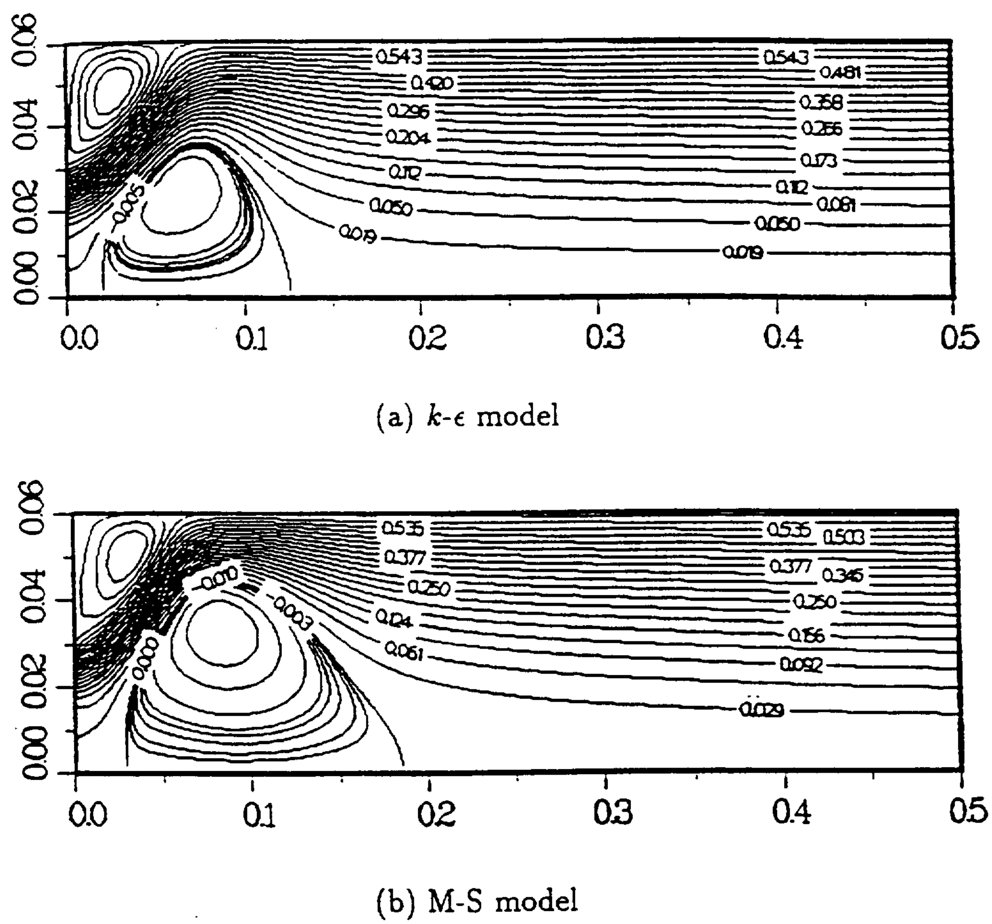
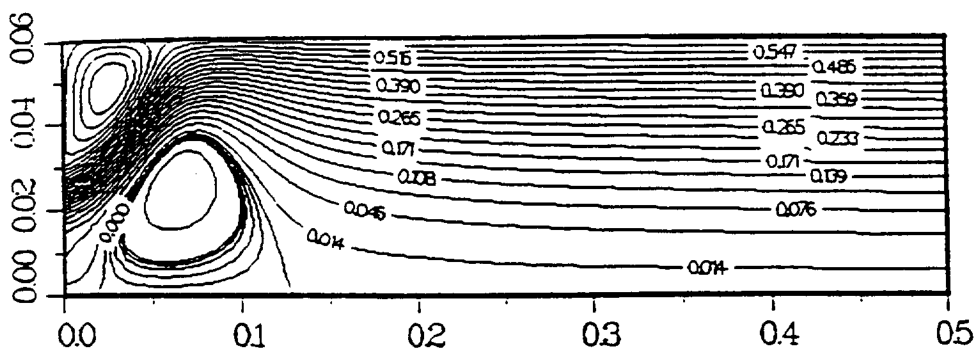
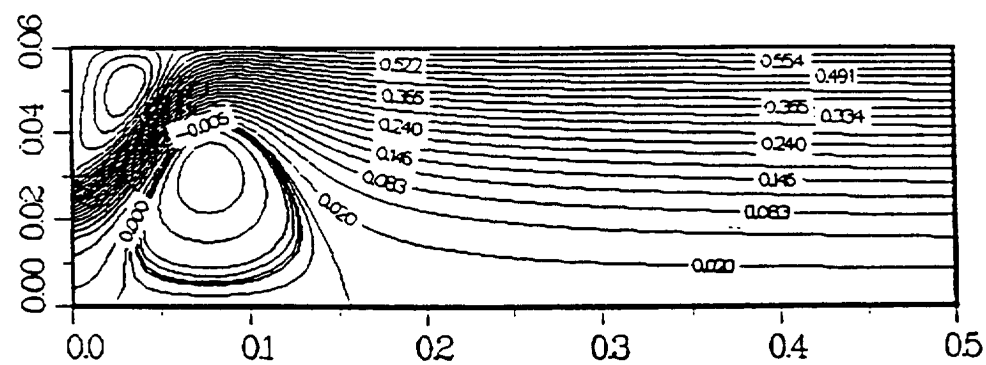


Figure 2. **Stream-function contours of confined swirling jet flow using the wall function near wall treatment**

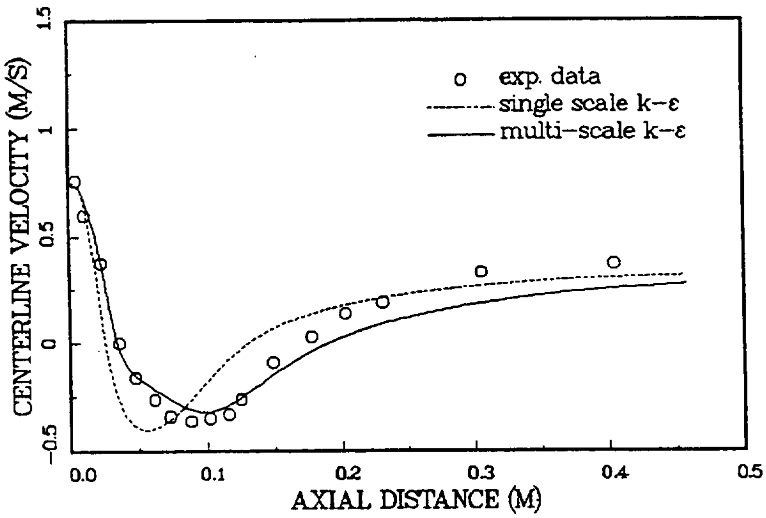


(a) k-ϵ model

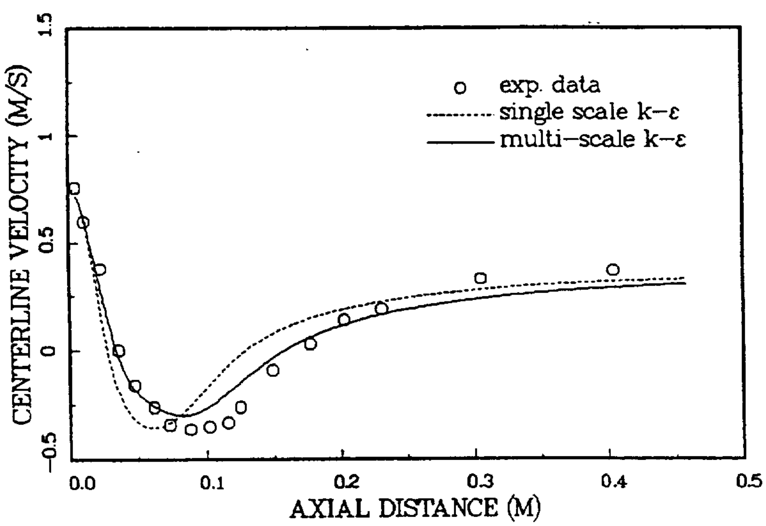


(b) M-S model

Figure 3. Stream-function contours of confined swirling jet flow using the two-layer near wall treatment



(a)



(b)

Figure 4. Axial mean velocity along the centerline
(a) wall function model
(b) two-layer model

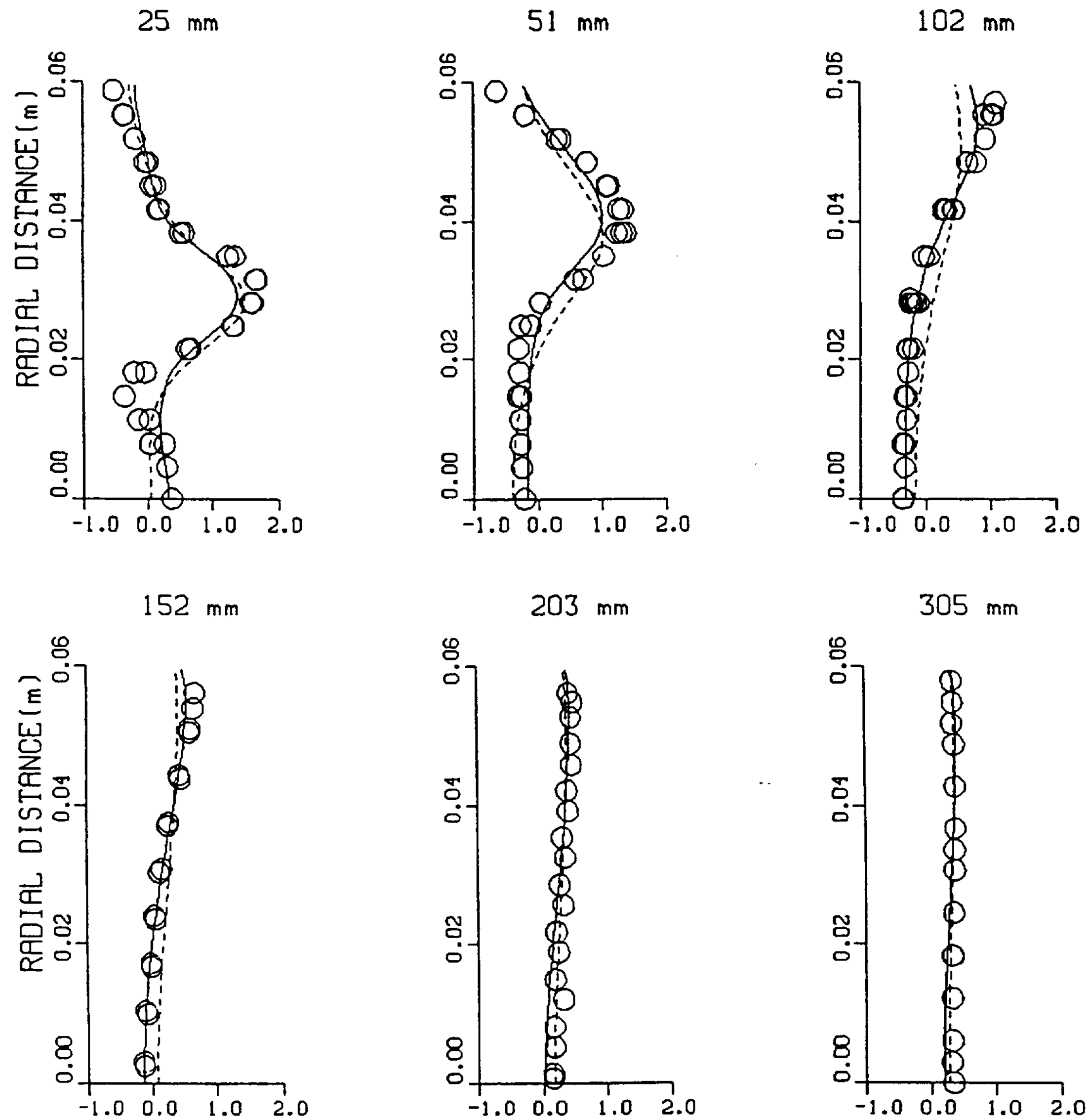


Figure 5. **Radial profiles of mean axial velocity using the wall function near wall treatment**

o exp. data
---- single scale *k*-*ε*
—— multi-scale *k*-*ε*

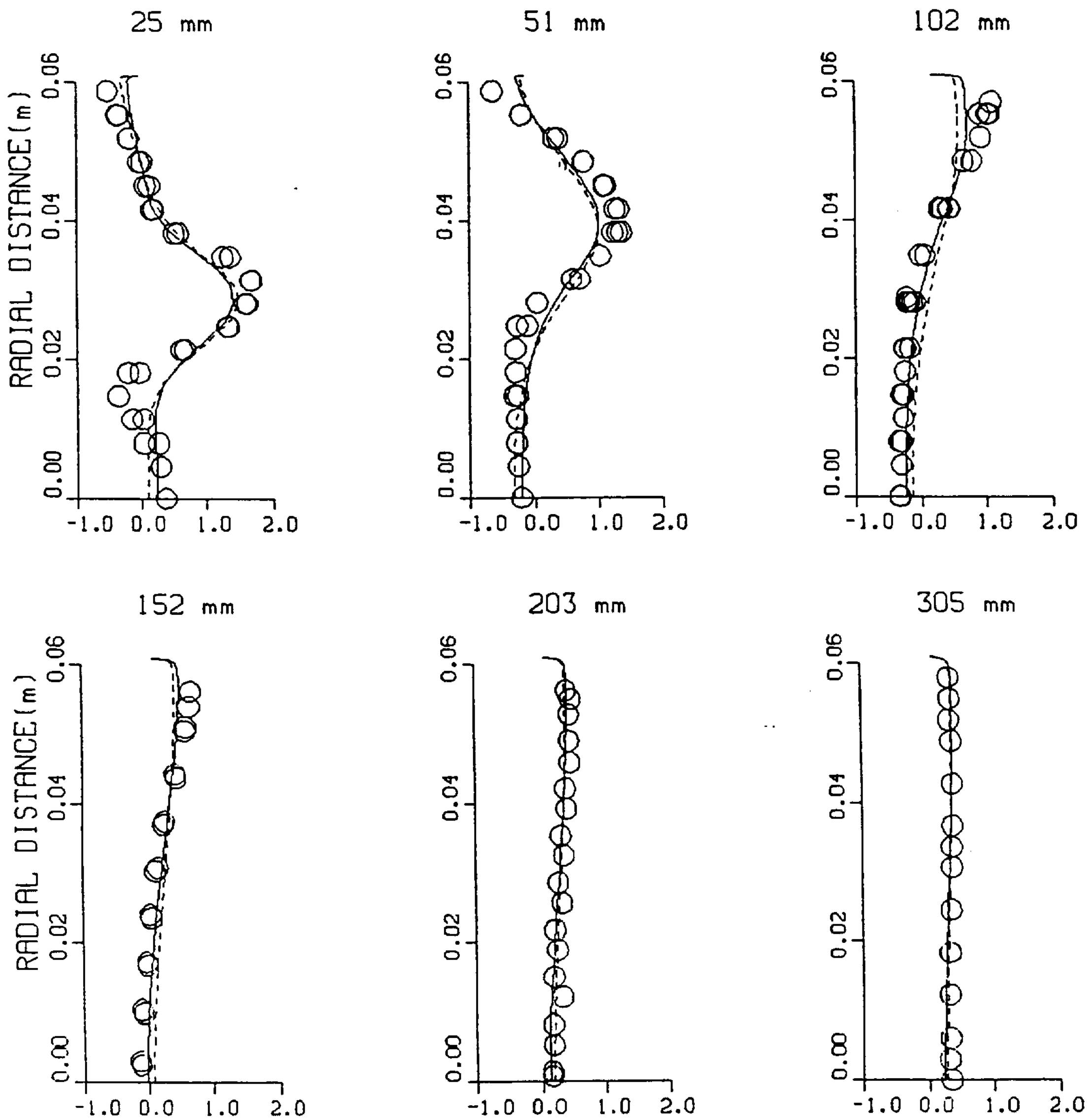


Figure 6. **Radial profiles of mean axial velocity using the two-layer near wall treatment**

o exp. data
---- single scale k-ε
—— multi-scale k-ε

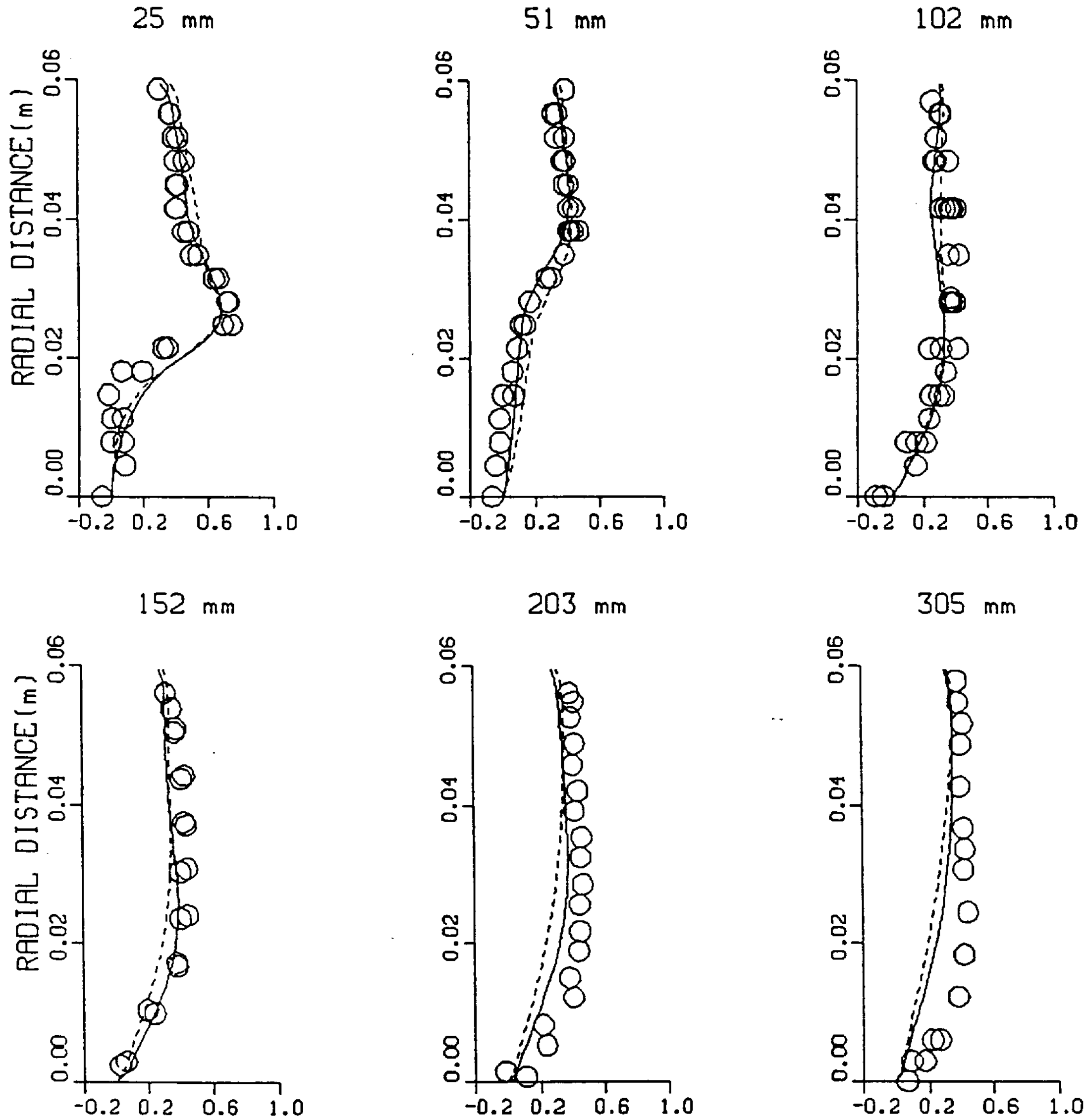


Figure 7 Radial profiles of mean tangential velocity using the wall function near wall treatment

o exp. data
---- single scale *k-ε*
—— multi-scale *k-ε*

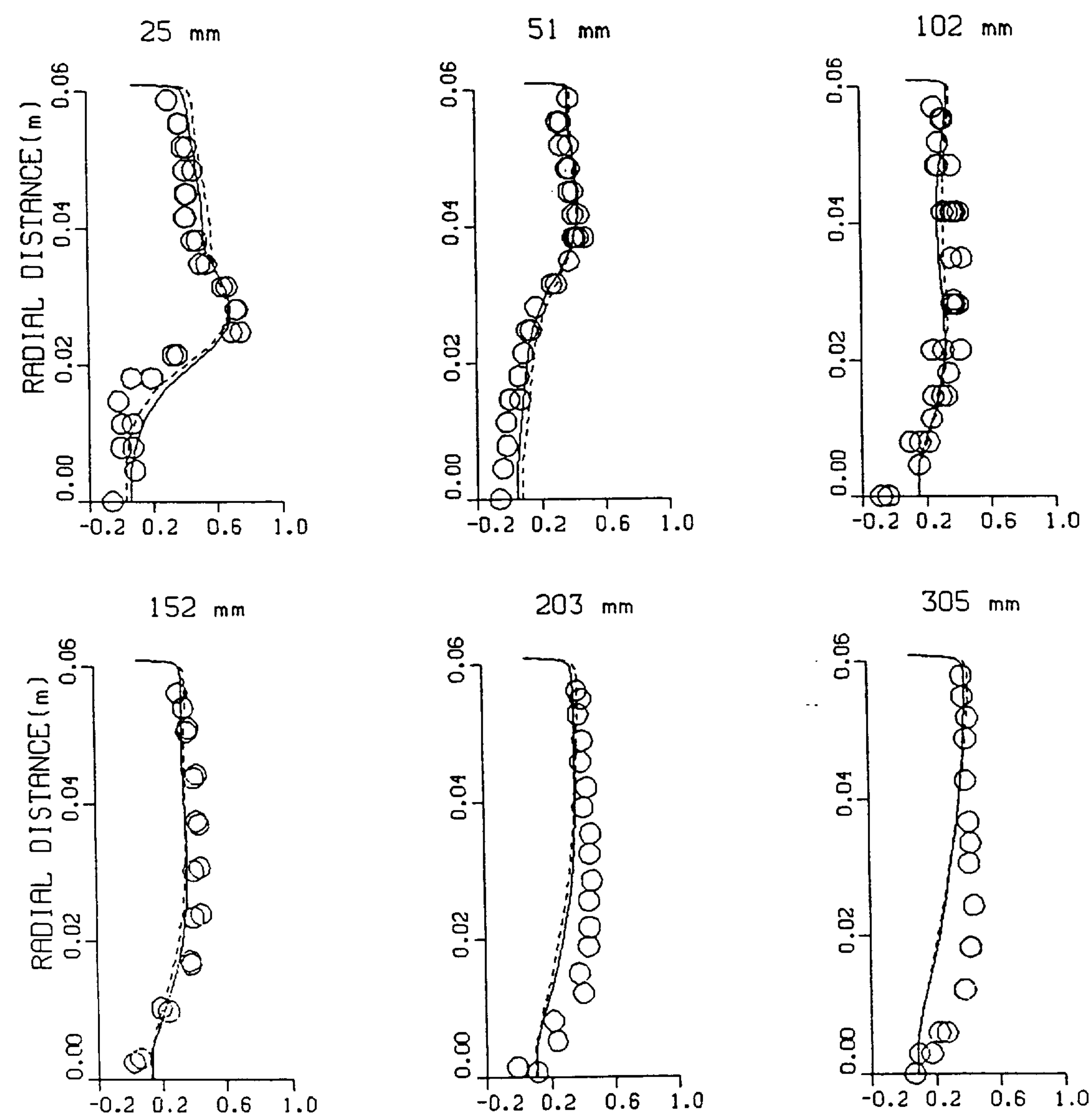


Figure 8. Radial profiles of mean tangential velocity using the two-layer near wall treatment

o exp. data
---- single scale k-ε
—— multi-scale k-ε

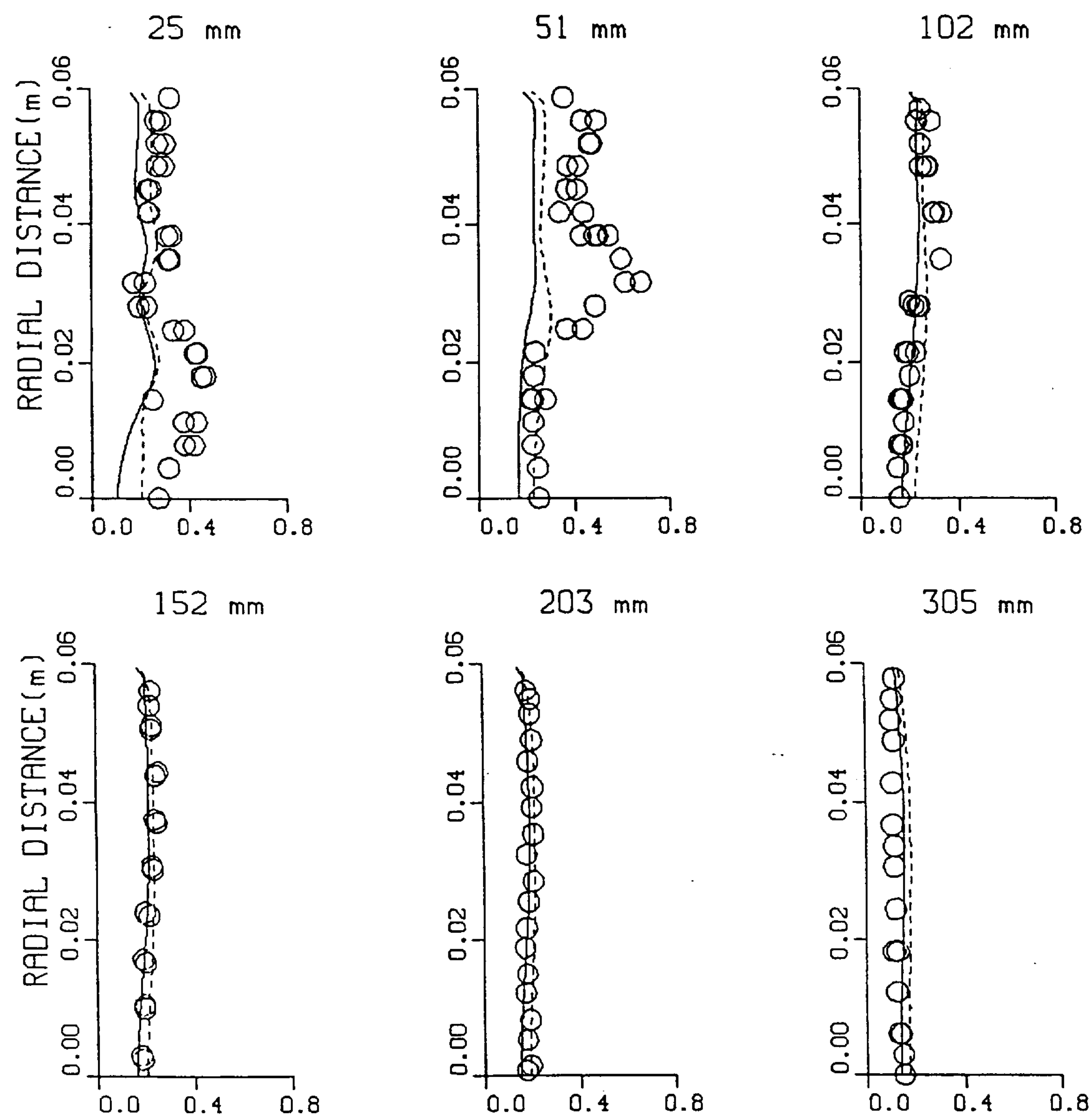


Figure 9. **Radial profiles of turbulent intensity $(\overline{uu})^{1/2}$ using the wall-function near wall treatment**

o exp. data
---- single scale k-ε
—— multi-scale k-ε

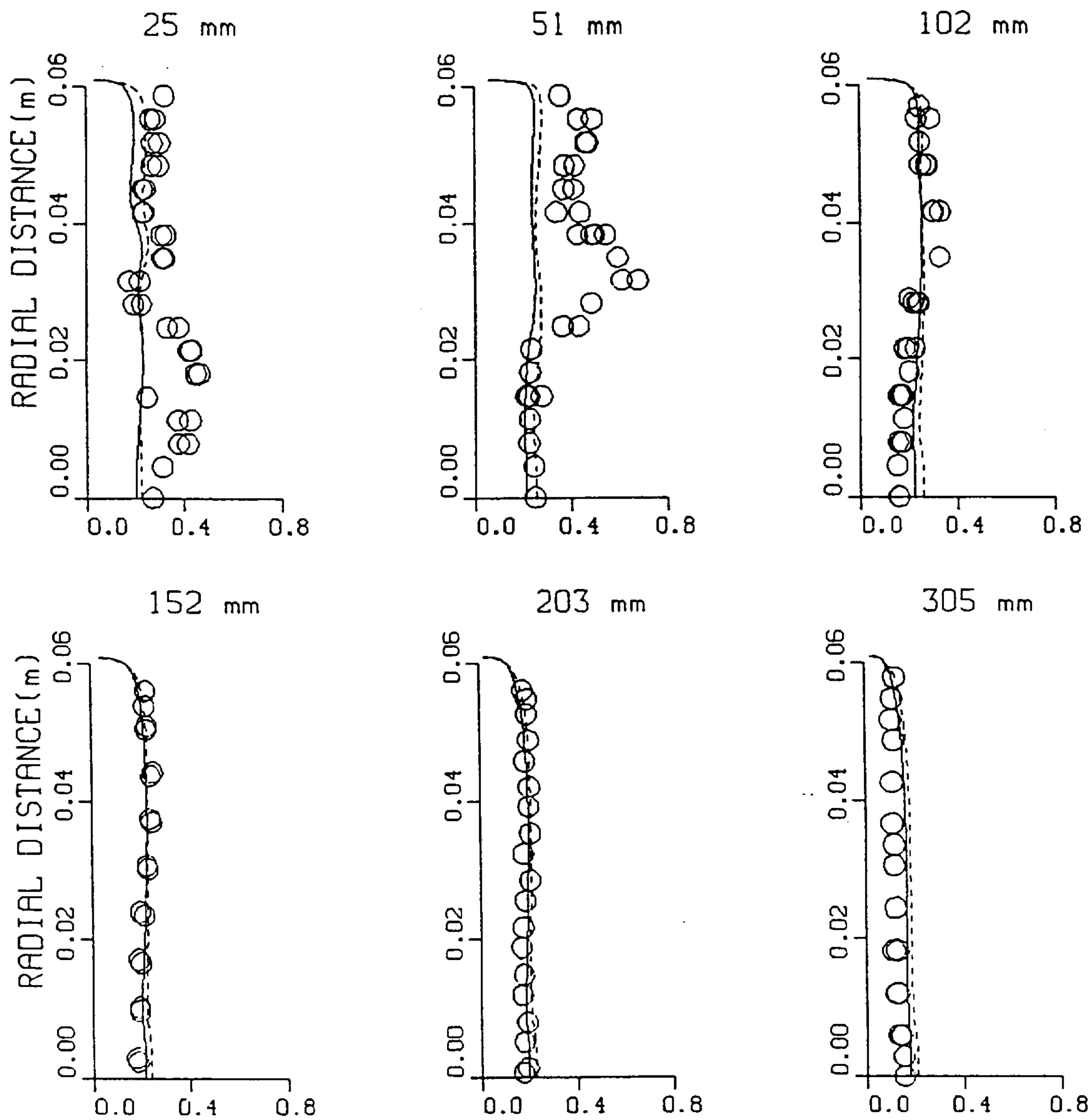


Figure 10. Radial profiles of turbulent intensity $(\overline{uu})^{1/2}$ using the two layer near-wall treatment

o exp. data
---- single scale k-ε
—— multi-scale k-ε

APPENDIX B

Multi-Scale k-ε Module Deck

B.1 Introduction

This user's manual describes the multi-scale k-ε module deck. The module is a self contained FORTRAN source code to compute turbulent kinetic energy, energy dissipation and turbulent eddy viscosity using the multi-time-scale k-ε turbulence model. It uses as input the mean flow properties as computed by conventional CFD techniques. The module is constructed to be self-contained, stand alone and compatible with a number of CFD solvers. A discussion of the multi-time-scale k -ε module structure is given next together with flow charts to show how to interface the module with a number of flow solvers. A list of variable names used is also given.

B.2 Program KEMOD

This is basically the solver for the k and ε - transport equations in both the production and the dissipation regions of the energy spectrum. It reads through its argument list different variables from the calling flow solver. These variables are described below where, each variable name ends with either an (I) for Integer variable, (R) for Real variable or (L) for Logical variable.

The flow chart of the program is shown in Figure B.1. It shows the main operations performed by the code.

<u>List of Argument Variable Names</u>

XR Grid node locations in the x or ξ-direction, dimensioned to XR (NX,NY) (input from flow solver)

YR Grid node locations in the y or η-direction, dimensioned to YR (NX,NY) (input from flow solver)

UR	Axial or x-direction velocity (u), dimensioned as UR (NX,NY) (input from flow solver
VR	Radial or y-direction velocity (v), also dimensional as VR (NX,NY) (input from flow solver)
WR	Azimuthal velocity (w), dimensional WR (NX,NY) (input from flow solver)
TER	Large scale turbulence kinetic energy k_p, dimensioned TER (NX,NY) (calculated in KEMOD and returned to flow solver)
EDR	Large scale turbulent energy dissipation rate ε_p, dimensioned EDR (NX,NY) (calculated in KEMOD and returned to flow solver)
TETR	Small scale turbulence kinetic energy k_t, dimensioned TETR (NX,NY) (calculated in KEMOD and returned to flow solver)
EDTR	Small scale turbulent energy dissipation rate ε_t, dimensioned EDTR (NX,NY) (calculated in KEMOD and returned to flow solver)
DENR	Fluid density, dimensioned DENR (NX,NY)
URFKER	Under-relaxation factors dimensioned as URFKER(4) and specified as follows: URFKER(1) for large scale turbulent energy equation URFKER(2) for small scale turbulent energy equation URFKER(3) for large scale turbulent energy dissipation equation URFKER(4) for small scale turbulent energy dissipation equation
PRTKER	Prandtl/Schmidt numbers dimensioned as PRTKER(4) and specified as follows: PRTKER(1) for large scale turbulent energy equation PRTKER(2) for small scale turbulent energy equation PRTKER(3) for large scale turbulent energy dissipation equation PRTKER(4) for small scale turbulent energy dissipation equation
GR	= 1.0 if second order upwinding is desired

= 0.0 if first order upwinding is used (input from flow solver).

F1R	Mass flux variable at cell faces in x- or ξ-direction, dimensioned F1R (NX,NY) (input from flow solver)
F2R	Mass flux variable at cell faces in y or η-direction, dimensioned F2R (NX,NY) (input from flow solver)
ITERI	Iteration number (input from flow solver), this number must be equal to 1 for a restart case
VISCOSR	Dynamic viscosity (input from flow solver)
VISR	Eddy viscosity, dimensioned VISR (NX,NY) (calculated in KEMOD and returned to main solver)
URFVISR	Under-relaxation factor for total viscosity calculation
AKSIL	Logical variable for axisymmetric geometry (AKSIL=·TRUE·) or plain geometry (AKSIL=·FALSE·) (input from flow solver)
C1R	Turbulence model constant, C_1 (input from flow solver)
C2R	Turbulence model constant, C_2 (input from flow solver)
CMUR	Turbulence model constant, C_μ (input from flow solver)
NIMI	Number of cell nodes in the I- or ξ-coordinate lines. (input from flow solver)
NJMI	Number of cell nodes in the J- or η-coordinate lines. (input from flow solver)
JTBEI	Boundary condition flag along east boundary must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set JTBEI to NJ*2, and similarly for other boundaries, dimensioned JTBEI (NY) (input from flow solver)

JTBWI	Boundary condition flag along west boundary, dimensioned JTBWI (NY) (input from flow solver)
ITBNI	Boundary condition flag along north boundary, dimensioned ITBNI (NX) (input from flow solver)
ITBSI	Boundary condition flag along south boundary, dimensioned ITBSI (NY) (input from flow solver)

Program KEMOD is interfaced with the main flow solver by a call to KEMOD with its arguments. For iterative flow solvers KEMOD is called within the iteration sequence after the solution of the momentum equations where the mean velocities are passed to KEMOD. There are different flow solvers utilizing different schemes from staggered to nonstaggered grid arrangement and for nonorthogonal coordinate system there are at least three alternatives to the choice of the velocity components

i. Cartesian velocity components
ii. Contravariant velocity components
iii. Covariant velocity components

The Cartesian velocity components are the most widely used and have the advantage of simple formulation of the governing equations. Whatever the arrangement used, mass fluxes at cell faces are required and passed to KEMOD as F1R and F2R in both directions. The location of other variables such as k and ε are at the cell center or cell nodes.

The module starts by reassigning variable names passed to it from flow solver to names that are shared with the different subroutines of the module in an include file "mske.h". The user must set the values for NX and NY in mske.h greater than or equal to the maximum grid dimensions. Then a check is made if it is the first iteration in which case the grid file "GRIDG" is called -after passing the grid node locations XR & YR in KEMOD- in order to calculate grid related quantities which will be explained later. The need to call GRIDG can be waived if all the grid data are passed to the module. That is all the information about the grid such as interpolation factors FX and FY, cell areas (ARE) and volumes (VOL) and normal distances of first grid point from grid boundaries

(DNS from south boundary, DNN - from north boundary, DNW - from west boundary and DNE - from east boundary).

After this, two calls to subroutine CALCKE are made to calculate the large and small scale turbulent kinetic energies with the identifier IPHI=1 and 2 respectively). The large and small scale energy dissipation equations are solved next by calling subroutine CALCKE again with the identifiers IPHI=3 and 4 respectively. The effective viscosity is calculated next. At locations where ε is close to zero (i.e., $\leq 10^{-30}$) viscosity is set to zero. A provision is made for under relaxing changes in effective viscosity which may help to stabilize oscillations and improve convergence rate.

B.3 Subroutines

GRIDG

Before calling this subroutine, the coordinates of all grid nodes, defined in reference to a fixed Cartesian coordinate frame are read. Figure B.2 shows the position of cell and grid nodes.

This subroutine is called only once to calculate coordinates of grid nodes (intersection of grid lines) and geometrical properties of the grid (cell areas and volumes, interpolation factors, normal distances of near-boundary cell nodes from boundary). All variables including grid node coordinates are converted to one-dimensional arrays. These are formed by scanning the grid in J-direction (figure B.2) for I=1, and then repeating for all I's. The position of any node in one-dimensional array is therefore defined as;

$$IJ = (I,J) = (I-1) * NJ + J$$

the actual number of grid nodes is one row and one column less than for all cell nodes. For I = NI and J = NJ fictitious grid nodes are introduced which have the same coordinates as actual nodes on NI-1 in I-direction and NJ-1 in J-direction.

The subroutine then calculates interpolation factors which are associated with cell nodes and are used in the main program to calculate values of dependent variables at locations other than cell nodes (cell centers). Definition of these are given in Figure B.3. Cell areas and volumes are calculated next followed by calculations of normal distances of near-boundary nodes from all four outer boundaries.

CALCKE (PHI, IPHI)

This subroutine solves the linearized and discretized transport equations for the turbulent energies (k_p and k_t) and the energy dissipation (ε_p and ε_t). The two dummy parameters in the calling statement, PHI and IPHI, represent arrays containing dependent variables for which the equation is to be solved. the subroutine sets up the convective and diffusive coefficients over the entire field. Then it calculates the source terms for either k or ε transport equations. A call is made to entry MODMSKE in order to modify these sources and boundary coefficients to suit the particular problem.

The discretized equations have the form

$$A_p \Phi_p = \sum_{i=EWNS} A_i \Phi_i + S_\Phi$$

where the coefficients A_i ($i=E,W,N,S$ see figure B.3) contain both the convective and diffusive fluxes. these equations are assembled and solved by calling subroutine SOLSIP which is based on Stone's Strongly Implicit Solver [7].

SOLSIP

This subroutine solves the system of linear algebraic equations for k and ε using Stone's Implicit Procedure [7]. The array RES (IJ) is used to store residuals. The sum of absolute residuals "RESORP" calculated in the first pass through this part of the routine is used as a measure of convergence of the solution process as a whole and this value is stored in RESOR (IPHI). This variable RESOR (IPHI) is passed to the main solver and if desired can be normalized and compared with the maximum error allowed there. If necessary inner iterations counter L and the sum of absolute residuals RESORP are printed out to monitor the rate of convergence of k and ε solution. If the ratio RSM is greater than the maximum allowed for the variable in question, SOR (IPHI), and the number of inner iterations is smaller than a prescribed maximum, NSWP (IPHI), then the routine repeats the sequence of calculating the residuals, increment vectors and updating the dependent variable.

MODMSKE

This subroutine is called from CALCKE subroutine and sets the boundary conditions for k_p, k_t and ε_p, ε_t depending on which variable being called (IDIR = 1, 2, 3, and 4 for k_p, k_t, ε_p, and ε_t respectively). Consider the south boundary for example, if it is one of four options:

(1) An inflow boundary ITBS(I) = 1, where the source term is set to accept the inlet values at J = 1 (south boundary)

(2) Outflow boundary ITBS(I) = 2, where zero gradient in y or η-direction is employed.

(3) Symmetry boundary, TBS(I) = 3, where gradients normal to symmetry plane are zero.

(4) Wall boundary, ITBS(I) = 4, where the production term GENTS(I) calculated form subroutine WALLFN in program MODIFY is added to the rest of the source term SU(IJ).

B.4 Program MODIFY

This subroutine is called from the u and v solver routines. It basically updates the flux source term of the discretized momentum equation due to wall shear stresses. If the u-momentum equation for example is discretized in the form

$$A_p^* u_p = \sum_{i=EWNS} A_i u_i + S_u^*$$

where P, E, W, N, S are cell nodes as shown in Figure B.3, and A_p^* and A_i's contain convective and diffusive coefficients. S_u^* is the source term containing pressure gradients and cross-derivative diffusion terms and convective terms for second-order upwinding scheme. This source term is usually linearized as $S_u^* = S_u - B_p u_p$ The term B_p is usually moved to the left hand side of the equation and modifies the diagonal coefficient $A_p = A_p^* + B_p$, and the equation can be written as

$$A_p u_p = \sum_{i=EWNS} A_i u_i + S_u$$

Then S_u and B_p are passed to subroutine MODIFY where they are modified if a wall is present (e.g., ITBS(I) = 4 for south boundary).

List of Argument Variable Names

CMU	Turbulence model constant, C_μ (input from flow solver)
VISCOS	Dynamic viscosity (input from flow solver)
XX	Grid node locations in the x or ξ-direction, dimensioned to XX (NX*NY) (input from flow solver)
YY	Grid node locations in the y or η-direction, dimensioned to YY (NX*NY) (input from flow solver)
R	Grid node radius equal to 1 for non-axisymmetric and YY for axisymmetric, dimensioned to R (NX*NY) (input from flow solver)
DNS	Normal distance to south, dimensioned to DNS (NX*NY) (input from flow solver)
DNN	Normal distance to north, dimensioned to DNN (NX*NY) (input from flow solver)
DNW	Normal distance to west, dimensioned to DNW (NX*NY) (input from flow solver)
DNE	Normal distance to east, dimensioned to DNE (NX*NY) (input from flow solver)

U Axial or x-direction velocity (u), dimensioned as UR (NX*NY) (input from flow solver

V Radial or y-direction velocity (v), also dimensional as VR (NX*NY) (input from flow solver)

W Azimuthal velocity (w), dimensional WR (NX*NY) (input from flow solver)

DEN Fluid density, dimensional DEN (NX*NY) (input from flow solver)

TE Large scale turbulence kinetic energy k_p, dimensioned TE (NX*NY) (calculated in KEMOD and returned to flow solver)

TET Small scale turbulence kinetic energy k_t, dimensioned TET (NX*NY) (calculated in KEMOD and returned to flow solver)

SU Variable source term, dimensioned SU (NX*NY)

BP Constant source term, dimensioned BP (NX*NY)

AE Cell area, dimensioned to AE (NX*NY) (input from flow solver)

AW Cell area, dimensioned to AW (NX*NY) (input from flow solver)

AN Cell area, dimensioned to AN (NX*NY) (input from flow solver)

AS Cell area, dimensioned to AS (NX*NY) (input from flow solver)

SUVS,SPVS,SUWS,SPWS

Source terms at south boundary due to wall shear stress, all dimensioned to S##S (NX*NY) (returned to flow solver)

SUVN,SPVN,SUWN,SPWN

Source terms at north boundary due to wall shear stress, all dimensioned to S##N (NX*NY) (returned to flow solver)

SUVW,SPVW,SUWW,SPWW

Source terms at west boundary due to wall shear stress, all dimensioned to S##W (NX*NY) (returned to flow solver)

SUVE,SPVE,SUWE,SPWE

Source terms at east boundary due to wall shear stress, all dimensioned to S##E (NX*NY) (returned to flow solver)

GENTS,GENTN,GENTW,GENTEE

Generation terms at south, north, west, and east boundaries respectively due to moving walls, with GENTS(NX), GENTN(NX), GENTW(NY), and GENTEE(NY) (returned to flow solver)

NX — Maximum number of cell nodes in the I- or ξ-coordinate lines. (input from flow solver)

NY — Maximum number of cell nodes in the J- or η-coordinate lines. (input from flow solver)

NXNY — NX*NY

NIM — Number of cell nodes in the I- or ξ-coordinate lines. (input from flow solver)

NJM — Number of cell nodes in the J- or η-coordinate lines. (input from flow solver)

ITBS — Boundary condition flag along south boundary must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set ITBS to NI*2, and similarly for other boundaries, dimensioned ITBS (NX) (input from flow solver)

ITBN — Boundary condition flag along north boundary, dimensioned ITBN (NX) (input from flow solver)

JTBW — Boundary condition flag along west boundary, dimensioned ITBNI (NY) (input from flow solver)

JTBE — Boundary condition flag along east boundary, dimensioned JTBE (NY) (input from flow solver)

For an iterative flow solver using the finite-volume methodology. A typical interface and call to KEMOD from the main flow solver can be represented by a flow chart as shown in figure B.4.

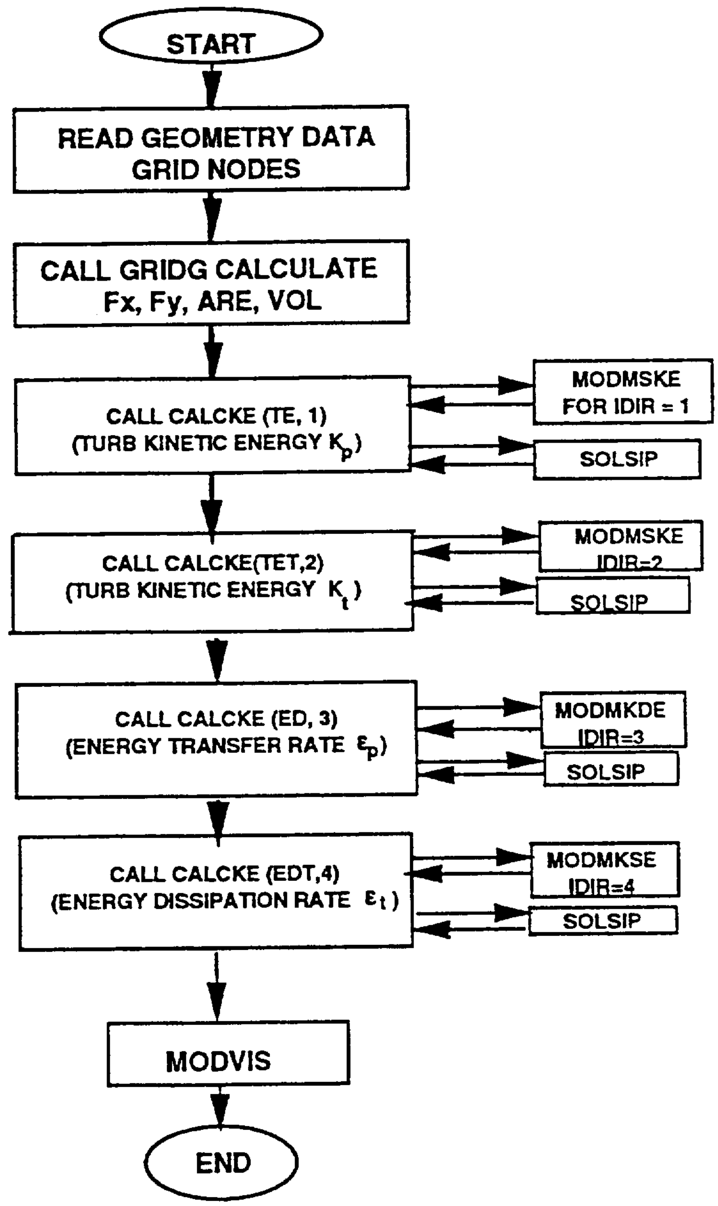


Figure B.1 Multi-scale k-ε module deck flow chart

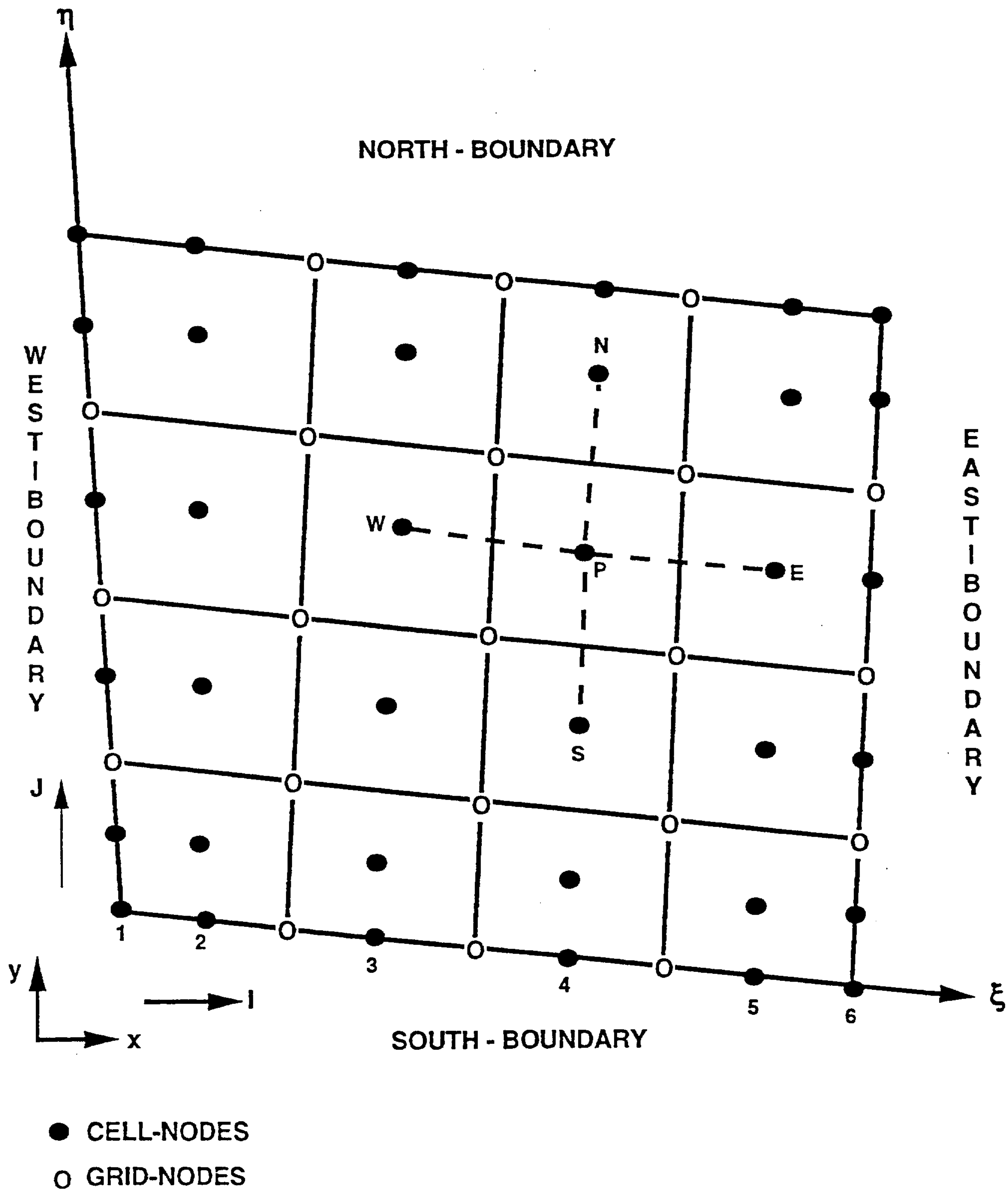
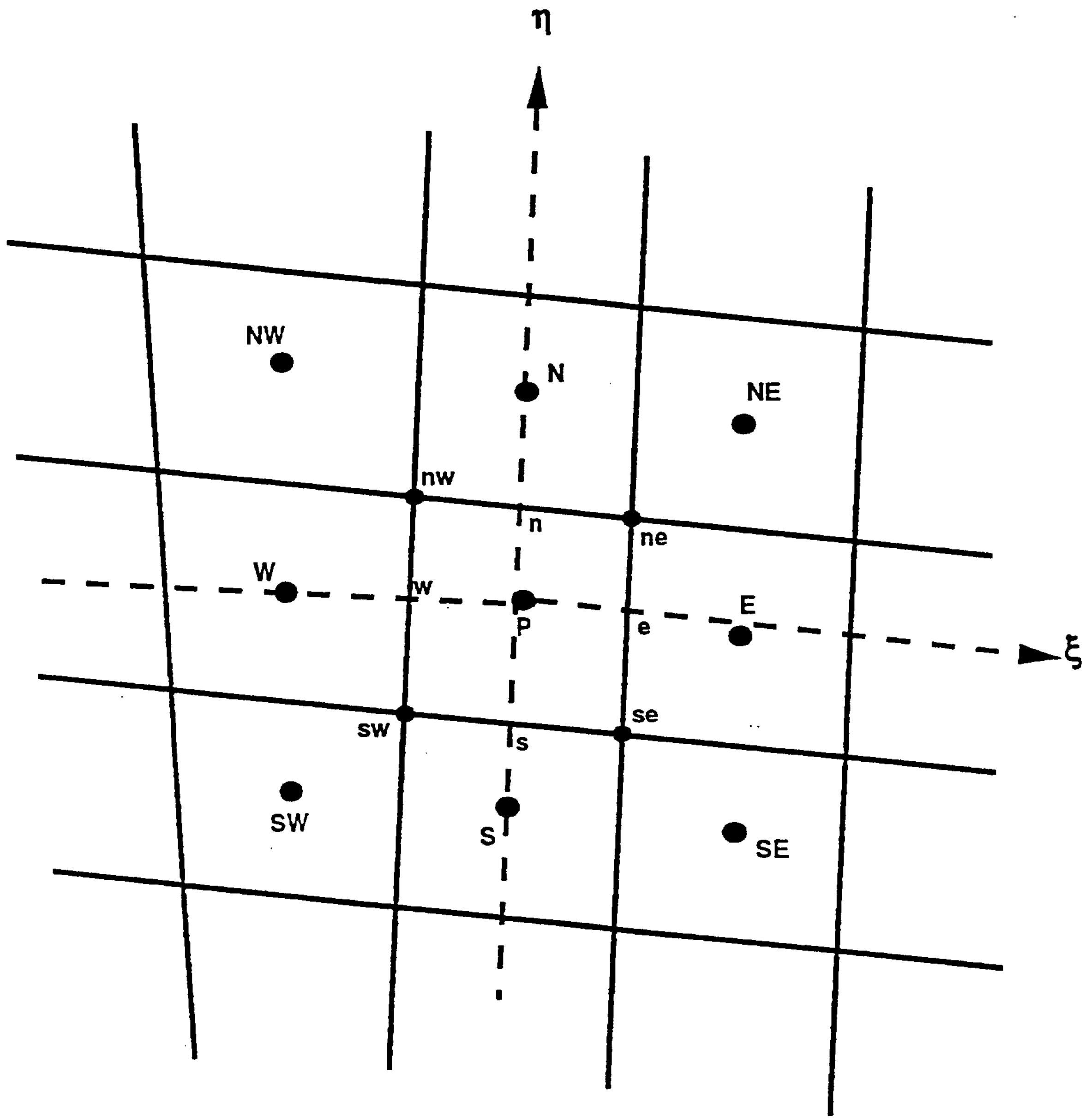


Figure B.2 Position of cell and grid nodes



$$FX_P = \frac{\overline{Pe}}{\overline{Pe} + \overline{eE}}, \quad FY_P = \frac{\overline{Pn}}{\overline{Pn} + \overline{nN}}$$

Figure B.3 Definition of interpolation factors

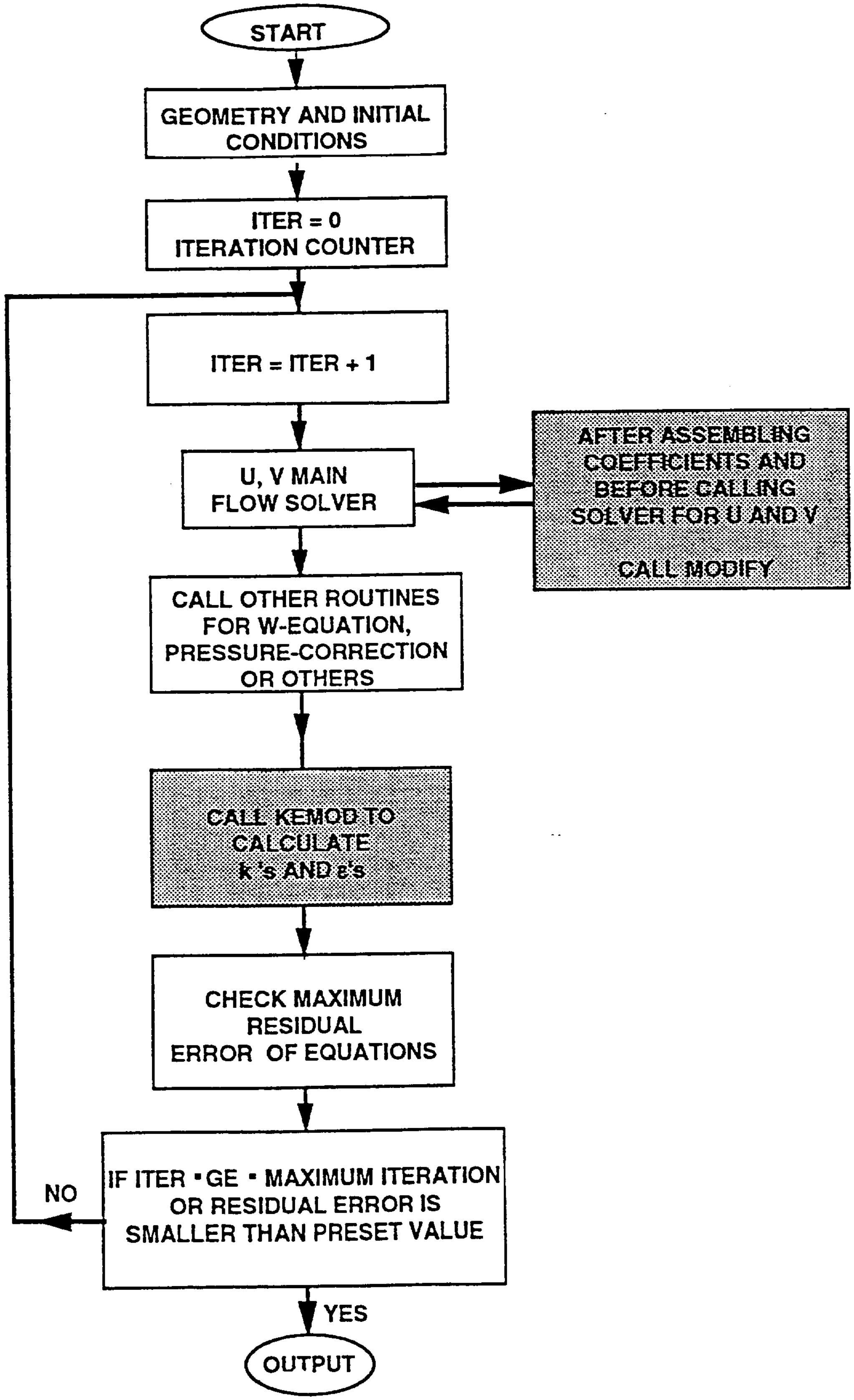


Figure B.4 Typical main flow solver flow chart with calls to the multi-scale k-e module

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                        C
C     2D/AXISYMMETRIC MULTI-SCALE K-E  TURBULENCE MODULE                 C
C                                                                        C
C     Rocketdyne CFD Technology Center                                   C
C                                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  Multi-Scale 2-Equation with 2 Wall Treatments
C  1.) Wall Function  2.) Two Layer
C
C-----------------------------------------------------------
      SUBROUTINE KEMOD (NIMI,NJMI,XR,YR,UR,VR,WR,TER,TETR,
     & EDR,EDTR,URFKPR,URFKTR,URFEDR,URFEDTR,PRTKPR,PRTKTR,
     & PRTEDR,PRTEDTR,GR,F1R,F2R,ITERI,VISCOSR,VISR,AKSIL,
     & LAY2L,CMUR,I2LWI,I2LEI,J2LSI,J2LNI,JTBEI,JTBWI,ITBNI,
     & ITBSI)
C-----------------------------------------------------------
      INCLUDE 'mskemod.h'
C
      DIMENSION XR(NX,NY),YR(NX,NY),UR(NX,NY),VR(NX,NY),
     & WR(NX,NY),TER(NX,NY),TETR(NX,NY),EDR(NX,NY),
     & EDTR(NX,NY),F1R(NX,NY),F2R(NX,NY),VISR(NX,NY),
     & JTBEI(NY),JTBWI(NY),ITBNI(NX),ITBSI(NX)
C
      COMMON/GR/ X(NX,NY),Y(NX,NY)
C
      DATA GREAT,SMALL/1.E30,1.E-30/
      DATA HAF,QTR/0.5,0.25/
      DATA SOR/0.1,0.1,0.1,0.1/
      DATA NSWP/10,10,10,10/
      DATA CP1,CP2,CP3/0.21,1.24,1.84/
      DATA CT1,CT2,CT3/0.29,1.28,1.66/
C
C--EQUATE VARIABLES IN ARGUMENT LIST TO THOSE IN COMMON BLOCK
C
      NIM=NIMI
      NJM=NJMI
      NI=NIM+1
      NJ=NJM+1
      URFKP=URFKPR
      URFKT=URFKTR
      URFED=URFEDR
      URFEDT=URFEDTR
      PRTKP=PRTKPR
      PRTKT=PRTKTR
      PRTED=PRTEDR
      PRTEDT=PRTEDTR
      G=GR
      ITER=ITERI
      VISCOS=VISCOSR
      CMU=CMUR
C
C--LOGICAL VARIABLES
C
      AKSI=AKSIL
      LAY2=LAY2L
C
C--TWO-LAYER TURBULENT PARAMETRS
C
      I2LW=I2LWI
      I2LE=I2LEI
      J2LS=J2LSI
      J2LN=J2LNI
C
      DO 10 I=1,NI
      IMNJ(I)=(I-1)*NJ
10    CONTINUE
C
C--BOUNDARY CONDITION IDENTIFIERS
C
```



```
      DO 20 I=1,NIM
      ITBN(I)=ITBNI(I)
      ITBS(I)=ITBSI(I)
20    CONTINUE
C
      DO 30 J=1,NJM
      JTBE(J)=JTBEI(J)
      JTBW(J)=JTBWI(J)
30    CONTINUE
C
C
      DO 50 I=1,NI
      DO 50 J=1,NJ
         IJ=(I-1)*NJ+J
         X(I,J)=XR(I,J)
         Y(I,J)=YR(I,J)
         U(IJ)=UR(I,J)
         V(IJ)=VR(I,J)
         W(IJ)=WR(I,J)
         TE(IJ)=TER(I,J)
         TET(IJ)=TETR(I,J)
         ED(IJ)=EDR(I,J)
         EDT(IJ)=EDTR(I,J)
         VIS(IJ)=VISR(I,J)
         F1(IJ)=F1R(I,J)
         F2(IJ)=F2R(I,J)
50    CONTINUE
C
C
C-----------------------------------------
C---- CALCULATE GRID GEOMETRIC VARIABLES INITIALLY
C
      IF(ITER.LE.1) THEN
         CALL GRIDG
      ENDIF
C
C-----------------------------------------
C
C--- CALL KINETIC ENERGY SOLVER
      CALL CALCE (TE,1)
      CALL CALCE (TET,2)
C
C--------2-LAYER MODEL
      IF(LAY2) CALL TWOLAY
C
C--- CALL ENERGY DISSIPATION SOLVER
      CALL CALCE (ED,3)
      CALL CALCE (EDT,4)
C
C--------UPDATE AND CALCULATE EDDY VISCOSITY
      CALL MODVIS
C
      DO 60 I=1,NI
      DO 60 J=1,NJ
         IJ=(I-1)*NJ+J
         TER(I,J) = TE(IJ)
         TETR(I,J)= TET(IJ)
         EDR(I,J) = ED(IJ)
         EDTR(I,J)= EDT(IJ)
         VISR(I,J) = VIS(IJ)
 60   CONTINUE
C
      RETURN
      END
C
C---------------------------------------------
      SUBROUTINE GRIDG
C---------------------------------------------
      INCLUDE 'mskemod.h'
C
      COMMON/GR/ X(NX,NY),Y(NX,NY)
```

```
C
      NJ=NJM+1
      NI=NIM+1
      NINJ=NI*NJ
      DO 2 I=1,NI
      IMNJ(I)=(I-1)*NJ
2     CONTINUE
      DO 3 I=1,NIM
      J=NJM
      X(I,J+1)=X(I,J)
      Y(I,J+1)=Y(I,J)
3     CONTINUE
      DO 4 J=1,NJ
      I=NIM
      X(I+1,J)=X(I,J)
      Y(I+1,J)=Y(I,J)
4     CONTINUE
C
C... GRID ORIGIN AT X=0, Y=0
      DO 5 I=1,NI
      DO 5 J=1,NJ
      IJ=(I-1)*NJ+J
      XX(IJ)=X(I,J)
      YY(IJ)=Y(I,J)
5     CONTINUE
C
C-----CALCULATION OF INTERPOLATION FACTORS
C
      DO 6 IJ=1,NINJ
      FX(IJ)=0.
      FY(IJ)=0.
    6 CONTINUE
C
      DO 7 J=2,NJM
      IJ=J
      FX(IJ)=0.
      LIE=NIM-1
      DO 8 I=2,LIE
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IJM=IJ-1
      IMJ=IJ-NJ
      DXP=0.5*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXPE=0.5*(XX(IPJ)-XX(IJ)+XX(IPJ-1)-XX(IJM))
      DYP=0.5*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYPE=0.5*(YY(IPJ)-YY(IJ)+YY(IPJ-1)-YY(IJM))
      DKP=SQRT(DXP**2+DYP**2)
      DKE=SQRT(DXPE**2+DYPE**2)
      FX(IJ)=DKP/(DKP+DKE)
    8 CONTINUE
      IJ=IMNJ(NIM)+J
      FX(IJ)=1.0
    7 CONTINUE
C
      DO 9 I=1,NI
      IJ=IMNJ(I)+1
      FX(IJ)=FX(IJ+1)
      IJ=IMNJ(I)+NJ
      FX(IJ)=FX(IJ-1)
    9 CONTINUE
C
      DO 10 I=2,NIM
      IJ=IMNJ(I)+1
      FY(IJ)=0.0
      LJN=NJM-1
      DO 11 J=2,LJN
      IJ=IMNJ(I)+J
      IJP=IJ+1
      IJM=IJ-1
      IMJ=IJ-NJ
      DXP=0.5*(XX(IJ)+XX(IMJ)-XX(IJM)-XX(IMJ-1))
```



```
      DXPN=0.5*(XX(IJP)+XX(IMJ+1)-XX(IJ)-XX(IMJ))
      DYP=0.5*(YY(IJ)+YY(IMJ)-YY(IJM)-YY(IMJ-1))
      DYPN=0.5*(YY(IJP)+YY(IMJ+1)-YY(IJ)-YY(IMJ))
      DEP=SQRT(DXP**2+DYP**2)
      DEPN=SQRT(DXPN**2+DYPN**2)
      FY(IJ)=DEP/(DEP+DEPN)
   11 CONTINUE
      IJ=IMNJ(I)+NJM
      FY(IJ)=1.0
   10 CONTINUE
C
      DO 12 J=1,NJ
      FY(J)=FY(J+NJ)
      IJ=IMNJ(NI)+J
      FY(IJ)=FY(IJ-NJ)
   12 CONTINUE
C
C-----CALCULATION OF CELL AREAS
C
      DO 13 IJ=1,NINJ
      ARE(IJ)=0.
   13 CONTINUE
C
      DO 14 I=2,NIM
      DO 14 J=2,NJM
      IJ=IMNJ(I)+J
      DXNESW=XX(IJ)-XX(IJ-NJ-1)
      DYNESW=YY(IJ)-YY(IJ-NJ-1)
      DXNWSE=XX(IJ-NJ)-XX(IJ-1)
      DYNWSE=YY(IJ-NJ)-YY(IJ-1)
      ARE(IJ)=0.5*ABS(DXNESW*DYNWSE-DXNWSE*DYNESW)
   14 CONTINUE
C
C-----NORMAL DISTANCE FROM THE WALL
C
      DO 15 I=1,NI
      DNS(I)=0.
      DNN(I)=0.
15    CONTINUE
C
      DO 16 J=1,NJ
      DNW(J)=0.
      DNE(J)=0.
   16 CONTINUE
C
      DO 17 I=2,NIM
      IJ=IMNJ(I)+2
      IMJ=IJ-NJ
      DXB=XX(IJ-1)-XX(IMJ-1)
      DYB=YY(IJ-1)-YY(IMJ-1)
      DXBP=0.25*(XX(IJ)-XX(IJ-1)+XX(IMJ)-XX(IMJ-1))
      DYBP=0.25*(YY(IJ)-YY(IJ-1)+YY(IMJ)-YY(IMJ-1))
      DNS(I)=DELTA(DXB,DYB,DXBP,DYBP)
      IJ=IMNJ(I)+NJM
      IMJ=IJ-NJ
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=0.25*(XX(IJ-1)-XX(IJ)+XX(IMJ-1)-XX(IMJ))
      DYBP=0.25*(YY(IJ-1)-YY(IJ)+YY(IMJ-1)-YY(IMJ))
      DNN(I)=DELTA(DXB,DYB,DXBP,DYBP)
   17 CONTINUE
C
      DO 18 J=2,NJM
      IJ=IMNJ(2)+J
      IMJ=IJ-NJ
      DXB=XX(IMJ)-XX(IMJ-1)
      DYB=YY(IMJ)-YY(IMJ-1)
      DXBP=0.25*(XX(IJ)-XX(IMJ)+XX(IJ-1)-XX(IMJ-1))
      DYBP=0.25*(YY(IJ)-YY(IMJ)+YY(IJ-1)-YY(IMJ-1))
      DNW(J)=DELTA(DXB,DYB,DXBP,DYBP)
      IJ=IMNJ(NIM)+J
```

```
      IMJ=IJ-NJ
      DXB=XX(IJ)-XX(IJ-1)
      DYB=YY(IJ)-YY(IJ-1)
      DXBP=0.25*(XX(IMJ)-XX(IJ)+XX(IMJ-1)-XX(IJ-1))
      DYBP=0.25*(YY(IMJ)-YY(IJ)+YY(IMJ-1)-YY(IJ-1))
      DNE(J)=DELTA(DXB,DYB,DXBP,DYBP)
   18 CONTINUE
C
C---- CALCULATE CELL VOLUMES
C
      DO 19 IJ=1,NINJ
      VOL(IJ)=ARE(IJ)
   19 CONTINUE
C
      IF(AKSI) THEN
      SIXR=1./6.
      DO 20 I=2,NIM
      II=IMNJ(I)
      DO 21 J=2,NJM
      IJ=II+J
      IMJ=IJ-NJ
      IMJM=IMJ-1
      IJM=IJ-1
      RIJ=YY(IJ)**2
      RIMJ=YY(IMJ)**2
      RIMJM=YY(IMJM)**2
      RIJM=YY(IJM)**2
      VOL(IJ)=SIXR*((XX(IJ)-XX(IMJ))*(RIJ+RIMJ+YY(IJ)*YY(IMJ))+
     &        (XX(IMJ)-XX(IMJM))*(RIMJ+RIMJM+YY(IMJ)*YY(IMJM))+
     &        (XX(IMJM)-XX(IJM))*(RIMJM+RIJM+YY(IMJM)*YY(IJM))+
     &        (XX(IJM)-XX(IJ))*(RIJM+RIJ+YY(IJM)*YY(IJ)))
   21 CONTINUE
   20 CONTINUE
      ENDIF
C
C---- INITIALIZE VARIABLES INITIALLY
C
      HAF=0.5
      QTR=0.25
      SMALL=1.E-30
      GREAT=1.E30
      DO 22 IJ=1,NINJ
      DEN(IJ)=DENSIT
      VIS(IJ)=VISCOS
      FMU(IJ)=1.0
      FLR1(IJ)=1.0
      FLR2(IJ)=1.0
      APU(IJ)=0.0
      APV(IJ)=0.0
      AE(IJ)=0.0
      AS(IJ)=0.0
      AN(IJ)=0.0
      AW(IJ)=0.0
      BE(IJ)=0.0
      BW(IJ)=0.0
      BN(IJ)=0.0
      BS(IJ)=0.0
      RES(IJ)=0.0
      R(IJ)=1.0
22    CONTINUE
      IF(AKSI) THEN
      DO 23 IJ=1,NINJ
      R(IJ)=YY(IJ)
23    CONTINUE
      ENDIF
C
      RETURN
      END
C
C
C*************************************************
```



```
      SUBROUTINE CALCE (PHI,IPHI)
C*************************************************
C
      INCLUDE 'mskemod.h'
C
      DIMENSION UW(NY),VW(NY),WW(NY),PHI(NXNY),FXWW(NY),DW(NY)
C
      GO TO (1,2,3,4) IPHI
1     CONTINUE
      URPHI=URFKP
      PRTINVP=1./PRTKP
      GO TO 5
2     CONTINUE
      URPHI=URFKT
      PRTINVP=1./PRTKT
      GO TO 5
3     CONTINUE
      URPHI=URFED
      PRTINVP=1./PRTED
      GO TO 5
4     CONTINUE
      URPHI=URFEDT
      PRTINVP=1./PRTEDT
5     CONTINUE
C
      IJ=1
      PHINE=PHI(IJ)
      PHINW(IJ)=PHINE
C
      DO 10 J=2,NJM
      IJ=J
      IJM=IJ-1
      IPJ=IJ+NJ
      IJP=IJ+1
      FYN=FY(IJ)
      FYS=1.0-FYN
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      DXE=XX(IJ)-XX(IJM)
      DYE=YY(IJ)-YY(IJM)
      VIST=VIS(IJ)-VISCOS
      GAME=HAF*(VISCOS+VIST*PRTINV(IPHI))*(R(IJ)+R(IJM))
      DW(J)=GAME/AREE*(DXE**2+DYE**2)
      PHISE=PHINE
      PHINE=PHI(IJ+1)*FYN+PHI(IJ)*FYS
      PHINW(J)=PHINE
      UW(J)=U(IJ)
      VW(J)=V(IJ)
      WW(J)=W(IJ)
      SNSW(J)=0.
      FXWW(J)=1.0
      IF(JTBW(J).EQ.3.OR.JTBW(J).EQ.4) GO TO 10
      DXKS=QTR*(XX(IPJ)+XX(IPJ-1)-XX(IJ)-XX(IJM))
      DYKS=QTR*(YY(IPJ)+YY(IPJ-1)-YY(IJ)-YY(IJM))
      SNSW(J)=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
   10 CONTINUE
C
      DO 100 I=2,NIM
      J=1
      IJ=IMNJ(I)+J
      IMJ=IJ-NJ
      IJP=IJ+1
      FXE=FX(IJ)
      FXW=1.-FX(IJ)
      AREN=HAF*(ARE(IJ)+ARE(IJP))
      DXN=XX(IJ)-XX(IMJ)
      DYN=YY(IJ)-YY(IMJ)
      VIST=VIS(IJ)-VISCOS
      GAMN=HAF*(VISCOS+VIST*PRTINV(IPHI))*(R(IJ)+R(IMJ))
      DN=GAMN/AREN*(DXN**2+DYN**2)
      FYSS=1.0
      PHINE=PHI(IJ+NJ)*FXE+PHI(IJ)*FXW
```

```
      UN=U(IJ)
      VN=V(IJ)
      WN=W(IJ)
      SEWN=0.
      IF(ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 110
      DXET=QTR*(XX(IJP)+XX(IJP-NJ)-XX(IJ)-XX(IMJ))
      DYET=QTR*(YY(IJP)+YY(IJP-NJ)-YY(IJ)-YY(IMJ))
      SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
      SEWNBS=SEWN
  110 CONTINUE
      PHINW(J)=PHINE
C
C-----THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES
C
      DO 101 J=2,NJM
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      IJP=IJ+1
      IJM=IJ-1
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.-FYN
      DXE=XX(IJ)-XX(IJM)
      DYE=YY(IJ)-YY(IJM)
      DXN=XX(IJ)-XX(IMJ)
      DYN=YY(IJ)-YY(IMJ)
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      AREN=HAF*(ARE(IJ)+ARE(IJP))
      VISE=VIS(IJ)*FXW+VIS(IPJ)*FXE
      VISET=VISE-VISCOS
      GAME=HAF*(VISCOS+VISET*PRTINV(IPHI))*(R(IJ)+R(IJM))
      VISN=VIS(IJ)*FYS+VIS(IJP)*FYN
      VISNT=VISN-VISCOS
      GAMN=HAF*(VISCOS+VISNT*PRTINV(IPHI))*(R(IJ)+R(IMJ))
C
      DS=DN
      DE=GAME/AREE*(DXE**2+DYE**2)
      DN=GAMN/AREN*(DXN**2+DYN**2)
C
C LINEAR UPWIND DIFFERENCING
C
      AEE=MIN(F1(IJ),0.0)*FX(IPJ)*G
      AWW=-MAX(F1(IMJ),0.0)*(1.0-FXWW(J))*G
      AE1=-MIN(F1(IMJ),0.0)*FXE*G
      AW1=MAX(F1(IJ),0.0)*(1.0-FX(IMJ))*G
      ANN=MIN(F2(IJ),0.0)*FY(IJP)*G
      ASS=-MAX(F2(IJM),0.0)*(1.0-FYSS)*G
      AN1=-MIN(F2(IJM),0.0)*FYN*G
      AS1=MAX(F2(IJ),0.0)*(1.0-FY(IJM))*G
C
      AW(IJ)=DW(J)+MAX(F1(IMJ),0.0)-AWW
      AE(IJ)=DE-MIN(F1(IJ),0.0)-AEE
      AS(IJ)=DS+MAX(F2(IJM),0.0)-ASS
      AN(IJ)=DN-MIN(F2(IJ),0.0)-ANN
C
      DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))
      DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))
      DXET=QTR*(XX(IJP)-XX(IJM)+XX(IJP-NJ)-XX(IJM-NJ))
      DYET=QTR*(YY(IJP)-YY(IJM)+YY(IJP-NJ)-YY(IJM-NJ)+2.0*STAG)
C
      PHISE=PHINE
      PHINE=(PHI(IJ)*FYS+PHI(IJP)*FYN)*FXW+
     &      (PHI(IPJ)*FYS+PHI(IPJ+1)*FYN)*FXE
      SEWS=SEWN
      SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
      SNSE=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
      IF(I.EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.
      IF(J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.
      APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS
```



```
C
C LINEAR UPWIND DIFFERENCING
C
      IMJ1=IMJ-NJ
      IMJJ=MAX(1,IMJ1)
      APV(IJ)=APV(IJ)+AEE*PHI(IPJ+NJ)+AWW*PHI(IMJJ)+ANN
     &       *PHI(IJP+1)+ASS*PHI(IJM-1)
     &       +AE1*PHI(IPJ)+AW1*PHI(IMJ)+AN1*PHI(IJP)+AS1*PHI(IJM)
      APU(IJ)=AEE+AWW+ANN+ASS+AE1+AW1+AN1+AS1
C
      VISS=VIS(IJ)-VISCOS+SMALL
C
      GO TO (120,121,130,131) IPHI
C
C-----TURBULENT KINETIC ENERGY SOURCE TERMS
C
  120 CONTINUE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))
      US=UN
      VS=VN
      WS=WN
      UN=U(IJP)*FYN+U(IJ)*FYS
      VN=V(IJP)*FYN+V(IJ)*FYS
      WN=W(IJP)*FYN+W(IJ)*FYS
      UE=U(IPJ)*FXE+U(IJ)*FXW
      VE=V(IPJ)*FXE+V(IJ)*FXW
      WE=W(IPJ)*FXE+W(IJ)*FXW
      DUEW=UE-UW(J)
      DUNS=UN-US
      DVEW=VE-VW(J)
      DVNS=VN-VS
      DWEW=WE-WW(J)
      DWNS=WN-WS
      GTERM=(2.*((DUEW*DYNS-DUNS*DYEW)**2+
     &        (DVNS*DXEW-DVEW*DXNS)**2)+(DUNS*DXEW-DUEW*DXNS+
     &         DVEW*DYNS-DVNS*DYEW)**2)/(ARE(IJ)**2)
      IF (AKSI)
     & GTERM=GTERM+((DYNS*DWEW-DYEW*DWNS)**2+
     &              (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &              /(ARE(IJ)**2)
     &           +2.*(V(IJ)/RP)**2

      DIVUV=(DUEW*DYNS-DUNS*DYEW+DVNS*DXEW-DVEW*DXNS)/ARE(IJ)
      GTERM=GTERM-2./3.*DIVUV**2
      GTERM=MAX(GTERM,0.)
C
C    GENERATION OF KINETIC ENERGY
C
      GEN(IJ)=(VIS(IJ)-VISCOS)*GTERM
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
C
      BP(IJ)=APU(IJ)+ED(IJ)*DEN(IJ)*VOL(IJ)/(TE(IJ)+SMALL)
C
      GO TO 150
C
C-----DISSIP.OF TURB. KIN. ENERGY SOURCE TERMS
C
  130 CONTINUE
C
      SU(IJ)=APV(IJ)+(CP1*GEN(IJ)**2/DEN(IJ)+CP2*GEN(IJ)*ED(IJ))*
     &       VOL(IJ)/(TE(IJ)+SMALL)
      BP(IJ)=APU(IJ)+CP3*DEN(IJ)*ED(IJ)*VOL(IJ)/(TE(IJ)+SMALL)
C
      GO TO 150
C
C--- SMALLER SCALE KINETIC ENERGY
C
```

```
121   CONTINUE
      SU(IJ)=APV(IJ)+(ED(IJ)-EDT(IJ))*DEN(IJ)*VOL(IJ)
      BP(IJ)=APU(IJ)
      GO TO 150
C
C--- SMALLER SCALE ENERGY DISSIPATION
C
131   CONTINUE
C
      SU(IJ)=APV(IJ)+(CT1*ED(IJ)**2+CT2*ED(IJ)*EDT(IJ))*DEN(IJ)*
     &                VOL(IJ)/(TET(IJ)+SMALL)
      BP(IJ)=APU(IJ)+DEN(IJ)*CT3*EDT(IJ)*VOL(IJ)/(TET(IJ)+SMALL)
C
150   CONTINUE
      UW(J)=UE
      VW(J)=VE
      WW(J)=WE
      SNSW(J)=SNSE
      PHINW(J)=PHINE
      FYSS=FY(IJM)
      FXWW(J)=FX(IMJ)
      DW(J)=DE
C
  101 CONTINUE
  100 CONTINUE
C
C-----PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS
C
      IDIR=IPHI
      CALL MODPHI
C
      IF(IPHI.GE.3.AND.LAY2) THEN
      DO 300 I=2,NIM
      IF(ITBS(I).NE.4) GO TO 301
      DO 310 J=2,J2LS
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
310   CONTINUE
301   IF(ITBN(I).NE.4) GO TO 300
      DO 320 J=J2LN,NJM
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
320   CONTINUE
300   CONTINUE
C
      DO 400 J=2,NJM
      IF(JTBW(J).NE.4) GO TO 401
      DO 410 I=2,I2LW
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
410   CONTINUE
401   IF(JTBE(J).NE.4) GO TO 400
      DO 420 I=I2LE,NIM
      IJ=IMNJ(I)+J
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
420   CONTINUE
400   CONTINUE
      ENDIF
C
      DO 200 I=2,NIM
      DO 201 J=2,NJM
      IJ=IMNJ(I)+J
      AP(IJ)=AW(IJ)+AE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)
      AP(IJ)=AP(IJ)/URFPHI
      SU(IJ)=SU(IJ)+(1.-URFPHI)*AP(IJ)*PHI(IJ)
  201 CONTINUE
  200 CONTINUE
```



```
C
C-----SOLVING F.D. EQUATIONS
C
      CALL SOLSIP(PHI,IPHI)
C
      RETURN
      END
C
C---------------------------------------------------
      SUBROUTINE TWOLAY
C---------------------------------------------------
      INCLUDE 'mskemod.h'
C
      DATA CMU25,CMU75,CAPPA,ELOG/0.5477,0.1643,0.4197,9.0/
C
      C11=CAPPA/CMU75
      AED=2.0*C11
      AMU=70.0
C
C...ALONG THE SOUTH BOUNDARY
C
      DO 70 I=2,NIM
      IF(ITBS(I).NE.4) GO TO 70
      DISNS=GREAT
      DISNW=GREAT
      DISNE=GREAT
C
      DO 71 J=2,J2LS
      IJ=IMNJ(I)+J
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XPW=HAF*(XX(IJW)+XX(IMJW))
      YPW=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XPW
      DYBP=YBP-YPW
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK WEST BOUNDARY
       IF(JTBW(J).EQ.4) THEN
       IJW=J
       IMJW=IJW-1
       DXB=XX(IJW)-XX(IMJW)
       DYB=YY(IJW)-YY(IMJW)
       XB=HAF*(XX(IJW)+XX(IMJW))
       YB=HAF*(YY(IJW)+YY(IMJW))
       XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
       YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
       DXBP=XBP-XB
       DYBP=YBP-YB
       DISNW=DELTA(DXB,DYB,DXBP,DYBP)
       ENDIF
C--CHECK EAST BOUNDARY
       IF(JTBE(J).EQ.4) THEN
       IJW=IMNJ(NIM)+J
       IMJW=IJW-1
       DXB=XX(IJW)-XX(IMJW)
       DYB=YY(IJW)-YY(IMJW)
       XB=HAF*(XX(IJW)+XX(IMJW))
       YB=HAF*(YY(IJW)+YY(IMJW))
       XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
       YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
       DXBP=XBP-XB
       DYBP=YBP-YB
       DISNE=DELTA(DXB,DYB,DXBP,DYBP)
       ENDIF
C
       DISN=MIN(DISNS,DISNW,DISNE)
       TETOT=TE(IJ)+TET(IJ)
```

```
      RK=DISN*DEN(IJ)*SQRT(TETOT)/VISCOS
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      ED(IJ)=TETOT*SQRT(TETOT)/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TETOT)*ALMU
 71   CONTINUE
 70   CONTINUE
C
C...ALONG THE NORTH BOUNDARY
C
      DO 72 I=2,NIM
      IF(ITBN(I).NE.4) GO TO 72
      DISNN=GREAT
      DISNW=GREAT
      DISNE=GREAT
      DO 73 J=J2LN,NJM
      IJ=IMNJ(I)+J
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK WEST BOUNDARY
      IF(JTBW(J).EQ.4) THEN
      IJW=J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNW=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C--CHECK EAST BOUNDARY
      IF(JTBE(J).EQ.4) THEN
      IJW=IMNJ(NIM)+J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNE=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C
      DISN=MIN(DISNN,DISNW,DISNE)
      TETOT=TE(IJ)+TET(IJ)
      RK=DISN*DEN(IJ)*SQRT(TETOT)/VISCOS
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      ED(IJ)=TETOT*SQRT(TETOT)/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TETOT)*ALMU
 73   CONTINUE
 72   CONTINUE
C
C
C...ALONG THE WEST BOUNDARY
C
      DO 75 J=2,NJM
```



```
      IF(JTBW(J).NE.4) GO TO 75
      DISNW=GREAT
      DISNN=GREAT
      DISNS=GREAT
      DO 76 I=2,I2LW
      IJ=IMNJ(I)+J
      IJW=J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNW=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK NORTH BOUNDARY
      IF(ITBN(I).EQ.4) THEN
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C--CHECK SOUTH BOUNDARY
      IF(ITBS(I).EQ.4) THEN
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XPW=HAF*(XX(IJW)+XX(IMJW))
      YPW=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XPW
      DYBP=YBP-YPW
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C
      DISN=MIN(DISNW,DISNS,DISNN)
      TETOT=TE(IJ)+TET(IJ)
      RK=DISN*DEN(IJ)*SQRT(TETOT)/VISCOS
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      ED(IJ)=TETOT*SQRT(TETOT)/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TETOT)*ALMU
 76   CONTINUE
 75   CONTINUE
C
C...ALONG THE EAST BOUNDARY
C
      DO 78 J=2,NJM
      IF(JTBE(J).NE.4) GO TO 78
      DISNE=GREAT
      DISNN=GREAT
      DISNS=GREAT
      DO 79 I=I2LE,NIM
      IJ=IMNJ(I)+J
      IJW=IMNJ(NIM)+J
      IMJW=IJW-1
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
```

```
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNE=DELTA(DXB,DYB,DXBP,DYBP)
C--CHECK NORTH BOUNDARY
      IF(ITBN(I).EQ.4) THEN
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XB=HAF*(XX(IJW)+XX(IMJW))
      YB=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C--CHECK SOUTH BOUNDARY
      IF(ITBS(I).EQ.4) THEN
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=XX(IJW)-XX(IMJW)
      DYB=YY(IJW)-YY(IMJW)
      XPW=HAF*(XX(IJW)+XX(IMJW))
      YPW=HAF*(YY(IJW)+YY(IMJW))
      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
      DXBP=XBP-XPW
      DYBP=YBP-YPW
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
      ENDIF
C
      DISN=MIN(DISNE,DISNS,DISNN)
      TETOT=TE(IJ)+TET(IJ)
      RK=DISN*DEN(IJ)*SQRT(TETOT)/VISCOS
      ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
      ALED=C11*DISN*(1.0-EXP(-RK/AED))
      ED(IJ)=TETOT*SQRT(TETOT)/ALED
      VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TETOT)*ALMU
   79 CONTINUE
   78 CONTINUE
C
      RETURN
      END
C
      FUNCTION DELTA(DXB,DYB,DXBP,DYBP)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      DELP=DXB*DXBP+DYB*DYBP
      DELTA=SQRT(DXBP**2+DYBP**2-DELP**2)
      RETURN
      END
C
C----------------------------------------------------
      SUBROUTINE SOLSIP(PHI,IPHI)
C----------------------------------------------------
      INCLUDE 'mskemod.h'
C
      DIMENSION PHI(NXNY)
      ID=IPHI
      L=0
      DO 5 I=2,NIM
      DO 5 J=2,NJM
      IJ=IMNJ(I)+J
      API=1.0/AP(IJ)
      AP(IJ)=1.0
      AE(IJ)=AE(IJ)*API
      AW(IJ)=AW(IJ)*API
```



```
      AN(IJ)=AN(IJ)*API
      AS(IJ)=AS(IJ)*API
      SU(IJ)=SU(IJ)*API
    5 CONTINUE
C
      DO 10 I=2,NIM
      DO 11 J=2,NJM
      IJ=IMNJ(I)+J
      IJM=IJ-1
      IMJ=IJ-NJ
      BW(IJ)=-AW(IJ)/(1.+ALFA*BN(IMJ))
      BS(IJ)=-AS(IJ)/(1.+ALFA*BE(IJM))
      POM1=ALFA*BW(IJ)*BN(IMJ)
      POM2=ALFA*BS(IJ)*BE(IJM)
      BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IMJ)-BS(IJ)*BN(IJM)
      BN(IJ)=(-AN(IJ)-POM1)/(BP(IJ)+SMALL)
      BE(IJ)=(-AE(IJ)-POM2)/(BP(IJ)+SMALL)
   11 CONTINUE
   10 CONTINUE
      NSTP=NSWP(ID)
  100 CONTINUE
      L=L+1
      RESORP=0.
      DO 20 I=2,NIM
      DO 21 J=2,NJM
      IJ=IMNJ(I)+J
      RES(IJ)=AN(IJ)*PHI(IJ+1)+AS(IJ)*PHI(IJ-1)+AE(IJ)*PHI(IJ+NJ)+
     &        AW(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)
      RESORP=RESORP+ABS(RES(IJ))
      RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/
     &        (BP(IJ)+SMALL)
   21 CONTINUE
   20 CONTINUE
      IF(L.EQ.1) RESOR(ID)=RESORP
      RSM=SOR(ID)*RESOR(ID)
      DO 30 I=2,NIM
      II=NIM+2-I
      DO 31 J=2,NJM
      JJ=NJM+2-J
      IJ=IMNJ(II)+JJ
      RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)
      PHI(IJ)=PHI(IJ)+RES(IJ)
   31 CONTINUE
   30 CONTINUE
      IF(TEST) PRINT 1, L,RESORP
      IF(RESORP.GT.RSM.AND.L.LT.NSTP) GO TO 100
      IF(RESORP.GT.RSM.AND.L.GE.NSTP) WRITE(6,2)
    1 FORMAT(10X,I5,' SWEEP, RESOR =',E12.4)
    2 FORMAT(//,10X,' SIPSOL DID NOT CONVERGE  ',//)
      RETURN
      END
C
C----------------------------------------------------
      SUBROUTINE USERM
C----------------------------------------------------
      INCLUDE 'mskemod.h'
C
C
      DATA CAPPA/0.4197/
C
C-----------------
      ENTRY MODVIS
C-----------------
      DO 80 I=1,NI
      DO 80 J=1,NJ
      IJ=IMNJ(I)+J
      VISOLD=VIS(IJ)
      VIS(IJ)=VISCOS
      IF(ED(IJ).GT.SMALL)
     & VIS(IJ)=DEN(IJ)*(TE(IJ)+TET(IJ))*TE(IJ)*CMU/ED(IJ)+VISCOS
      VIS(IJ)=URFVIS*VIS(IJ)+(1.-URFVIS)*VISOLD
```

```
   80 CONTINUE
C
      IF(LAY2) THEN
      DO 81 I=2,NIM
      IF(ITBS(I).NE.4) GO TO 82
      DO 83 J=2,J2LS
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
 83   CONTINUE
 82   IF(ITBN(I).NE.4) GO TO 81
      DO 84 J=J2LN,NJM
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
 84   CONTINUE
 81   CONTINUE
      DO 85 J=2,NJM
      IF(JTBE(J).NE.4) GO TO 86
      DO 87 I=I2LE,NIM
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
 87   CONTINUE
 86   IF(JTBW(J).NE.4) GO TO 85
      DO 88 I=2,I2LW
      IJ=IMNJ(I)+J
      VIS(IJ)=VIS2(IJ)
 88   CONTINUE
 85   CONTINUE
C
      ENDIF
C
      RETURN

C-----------------------
      ENTRY MODPHI
C-----------------------
      GO TO (800,900,1000,1100) IDIR
C
C-----BOUNDARY CONDITIONS FOR TURBULENT KINETIC ENERGY (KP)
C
  800 CONTINUE
C-----SOUTH BOUNDARY
      DO 810 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (811,812,813,814) ITBS(I)
  811 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 815
  812 TE(IJ-1)=TE(IJ)
      GO TO 815
  813 CONTINUE
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
      TE(IJ)=TE(IJ+1)-DEL*FAC
      IJ=IJ+1
      GO TO 815
  814 CONTINUE
      IF(.NOT.LAY2) GEN(IJ)=GENTS(I)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
```



```
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      WN=W(IJP)*FYN+W(IJ)*FYS
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &              (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &             /(ARE(IJ)**2)
     &           +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  815 CONTINUE
      AS(IJ)=0.0
  810 CONTINUE
C
C-----NORTH BOUNDARY
      DO 820 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (821,822,823,824) ITBN(I)
  821 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 825
  822 TE(IJ+1)=TE(IJ)
      GO TO 825
  823 CONTINUE
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
      TE(IJ)=TE(IJ-1)-DEL*FAC
      IJ=IJ-1
      GO TO 825
  824 CONTINUE
      IF(.NOT.LAY2) GEN(IJ)=GENTN(I)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
```

```
      WN=W(IJP)*FYN+W(IJ)*FYS
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &                (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &                /(ARE(IJ)**2)
     &             +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  825 CONTINUE
      AN(IJ)=0.0
  820 CONTINUE
C-----WEST BOUNDARY
      DO 830 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (831,832,833,834) JTBW(J)
  831 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*TE(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 835
  832 TE(IJ-NJ)=TE(IJ)
      GO TO 835
  833 CONTINUE
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2
      TE(IJ)=TE(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GO TO 835
  834 CONTINUE
      IF(.NOT.LAY2) GEN(IJ)=GENTW(J)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      WN=W(IJP)*FYN+W(IJ)*FYS
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &                (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &                /(ARE(IJ)**2)
     &             +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  835 CONTINUE
```



```
      AW(IJ)=0.0
  830 CONTINUE
C-----EAST BOUNDARY
      DO 840 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (841,842,843,844) JTBE(J)
  841 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*TE(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 845
  842 TE(IJ+NJ)=TE(IJ)
      GO TO 845
  843 CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2
      TE(IJ)=TE(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 845
  844 CONTINUE
      IF(.NOT.LAY2) GEN(IJ)=GENTEE(J)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      IF(AKSI) THEN
      IMJ=IJ-NJ
      IJM=IJ-1
      IJP=IJ+1
      IPJ=IJ+NJ
      FYN=FY(IJ)
      FYS=1.0-FYN
      FXE=FX(IJ)
      FXW=1.0-FXE
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      WN=W(IJP)*FYN+W(IJ)*FYS
      WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
      WE=W(IPJ)*FXE+W(IJ)*FXW
      WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
      DWEW=WE-WW
      DWNS=WN-WS
      GEN(IJ)=GEN(IJ)+( ((DYNS*DWEW-DYEW*DWNS)**2+
     &                (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)
     &                /(ARE(IJ)**2)
     &             +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
      ENDIF
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  845 CONTINUE
      AE(IJ)=0.0
  840 CONTINUE
C
      RETURN
C
C-- BOUNDARY CONDITIONS FOR SMALL SCALE K (KT)
C
900   CONTINUE
C-----SOUTH BOUNDARY
      DO 910 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (911,912,913,914) ITBS(I)
  911 CONTINUE
```

```
      SU(IJ)=SU(IJ)+AS(IJ)*TET(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 915
912   TET(IJ-1)=TET(IJ)
      GO TO 915
913   CONTINUE
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TET(IJ)*(FXW1-FXE2)+TET(IPJ)*FXE1-TET(IMJ)*FXW2
      TET(IJ)=TET(IJ+1)-DEL*FAC
      IJ=IJ+1
      GO TO 915
914   CONTINUE
      SU(IJ)=FRACTW*TE(IJ)*GREAT
      BP(IJ)=GREAT
915   CONTINUE
      AS(IJ)=0.0
910   CONTINUE
C-----NORTH BOUNDARY
      DO 920 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (921,922,923,924) ITBN(I)
921   CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*TET(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 925
922   TET(IJ+1)=TET(IJ)
      GO TO 925
923   CONTINUE
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TET(IJ)*(FXW1-FXE2)+TET(IPJ)*FXE1-TET(IMJ)*FXW2
      TET(IJ)=TET(IJ-1)-DEL*FAC
      IJ=IJ-1
      GO TO 925
924   CONTINUE
      SU(IJ)=FRACTW*TE(IJ)*GREAT
      BP(IJ)=GREAT
925   CONTINUE
      AN(IJ)=0.0
920   CONTINUE
C-----WEST BOUNDARY
      DO 930 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (931,932,933,934) JTBW(J)
931   CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*TET(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 935
932   TET(IJ-NJ)=TET(IJ)
      GO TO 935
```



```
933   CONTINUE
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TET(IJ)*(FYS1-FYN2)+TET(IJP)*FYN1-TET(IJM)*FYS2
      TET(IJ)=TET(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GO TO 935
934   CONTINUE
      SU(IJ)=FRACTW*TE(IJ)*GREAT
      BP(IJ)=GREAT
935   CONTINUE
      AW(IJ)=0.0
930   CONTINUE
C-----EAST BOUNDARY
      DO 940 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (941,942,943,944) JTBE(J)
941   CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*TET(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 945
942   TET(IJ+NJ)=TET(IJ)
      GO TO 945
943   CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TET(IJ)*(FYS1-FYN2)+TET(IJP)*FYN1-TET(IJM)*FYS2
      TET(IJ)=TET(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 945
944   CONTINUE
      SU(IJ)=FRACTW*TE(IJ)*GREAT
      BP(IJ)=GREAT
945   CONTINUE
      AE(IJ)=0.0
940   CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR ENERGY TRANSFER RATE (ED).
C
1000  CONTINUE
C
      CMU25=SQRT(SQRT(CMU))
      CMU75=CMU25**3
C
C-----SOUTH BOUNDARY
      DO 1010 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1011,1012,1013,1014) ITBS(I)
1011  CONTINUE
```

```
      SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1015
 1012 ED(IJ-NJ)=ED(IJ)
      GO TO 1015
 1013 CONTINUE
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ+1)-DEL*FAC
      IJ=IJ+1
      GO TO 1015
 1014 CONTINUE
      TT=ABS(TE(IJ)+TET(IJ))
      SU(IJ)=GREAT*CMU75*TT*SQRT(TT)/(CAPPA*DNS(I))
      BP(IJ)=GREAT
 1015 CONTINUE
      AS(IJ)=0.0
 1010 CONTINUE
C-----NORTH BOUNDARY
      DO 1020 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1021,1022,1023,1024) ITBN(I)
 1021 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1025
 1022 ED(IJ+1)=ED(IJ)
      GO TO 1025
 1023 CONTINUE
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      ED(IJ)=ED(IJ-1)-DEL*FAC
      IJ=IJ-1
      GO TO 1025
 1024 CONTINUE
      TT=ABS(TE(IJ)+TET(IJ))
      SU(IJ)=GREAT*CMU75*TT*SQRT(TT)/(CAPPA*DNN(I))
      BP(IJ)=GREAT
 1025 CONTINUE
      AN(IJ)=0.0
 1020 CONTINUE
C-----WEST BOUNDARY
      DO 1030 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1031,1032,1033,1034) JTBW(J)
 1031 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1035
```



```
 1032 ED(IJ-NJ)=ED(IJ)
      GO TO 1035
 1033 CONTINUE
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GO TO 1035
 1034 CONTINUE
      TT=ABS(TE(IJ)+TET(IJ))
      SU(IJ)=GREAT*CMU75*TT*SQRT(TT)/(CAPPA*DNW(J))
      BP(IJ)=GREAT
 1035 CONTINUE
      AW(IJ)=0.0
 1030 CONTINUE
C-----EAST BOUNDARY
      DO 1040 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1041,1042,1043,1044) JTBE(J)
 1041 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1045
 1042 ED(IJ+NJ)=ED(IJ)
      GO TO 1045
 1043 CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      ED(IJ)=ED(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 1045
 1044 CONTINUE
      TT=ABS(TE(IJ)+TET(IJ))
      SU(IJ)=GREAT*CMU75*TT*SQRT(TT)/(CAPPA*DNE(J))
      BP(IJ)=GREAT
 1045 CONTINUE
      AE(IJ)=0.0
 1040 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR ENERGY DISSIPATION RATE (EDT).
C
 1100 CONTINUE
C
C-----SOUTH BOUNDARY
      DO 1110 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1111,1112,1113,1114) ITBS(I)
```

```
 1111 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*EDT(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1115
 1112 EDT(IJ-NJ)=EDT(IJ)
      GO TO 1115
 1113 CONTINUE
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
      DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=EDT(IJ)*(FXW1-FXE2)+EDT(IPJ)*FXE1-EDT(IMJ)*FXW2
      EDT(IJ)=EDT(IJ+1)-DEL*FAC
      IJ=IJ+1
      GO TO 1115
 1114 CONTINUE
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
 1115 CONTINUE
      AS(IJ)=0.0
 1110 CONTINUE
C-----NORTH BOUNDARY
      DO 1120 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1121,1122,1123,1124) ITBN(I)
 1121 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*EDT(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1125
 1122 EDT(IJ+1)=EDT(IJ)
      GO TO 1125
 1123 CONTINUE
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=XX(IJ)-XX(IMJ)
      DYB=YY(IJ)-YY(IMJ)
      DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
      DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=EDT(IJ)*(FXW1-FXE2)+EDT(IPJ)*FXE1-EDT(IMJ)*FXW2
      EDT(IJ)=EDT(IJ-1)-DEL*FAC
      IJ=IJ-1
      GO TO 1125
 1124 CONTINUE
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
 1125 CONTINUE
      AN(IJ)=0.0
 1120 CONTINUE
C-----WEST BOUNDARY
      DO 1130 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1131,1132,1133,1134) JTBW(J)
 1131 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*EDT(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1135
 1132 EDT(IJ-NJ)=EDT(IJ)
```



```
      GO TO 1135
 1133 CONTINUE
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=YY(IJ)-YY(IJM)
      DXB=XX(IJ)-XX(IJM)
      DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=EDT(IJ)*(FYS1-FYN2)+EDT(IJP)*FYN1-EDT(IJM)*FYS2
      EDT(IJ)=EDT(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GO TO 1135
 1134 CONTINUE
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
 1135 CONTINUE
      AW(IJ)=0.0
 1130 CONTINUE
C-----EAST BOUNDARY
      DO 1140 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1141,1142,1143,1144) JTBE(J)
 1141 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*EDT(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1145
 1142 EDT(IJ+NJ)=EDT(IJ)
      GO TO 1145
 1143 CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=XX(IJ)-XX(IJM)
      DYB=YY(IJ)-YY(IJM)
      DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))
      DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=EDT(IJ)*(FYS1-FYN2)+EDT(IJP)*FYN1-EDT(IJM)*FYS2
      EDT(IJ)=EDT(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 1145
 1144 CONTINUE
      SU(IJ)=GREAT*ED(IJ)
      BP(IJ)=GREAT
 1145 CONTINUE
      AE(IJ)=0.0
 1140 CONTINUE
C
      END
C
C--- SUBROUTINE 'MODIFY' TO MODIFY SHEAR STRESSES IN
C--- NEAR WALL GRID POINT
C-------------------------------------------------------
      SUBROUTINE MODIFY (SUR,BPR)
C-------------------------------------------------------
C
      INCLUDE 'mskemod.h'
      DIMENSION SUR(NXNY),BPR(NXNY)
C
      DATA CMU25,CMU75,CAPPA,ELOG/0.5477,0.1643,0.4197,9.0/
C
```

```
      DO 10 I=1,NI
      DO 10 J=1,NJ
      IJ=IMNJ(I)+J
      SU(IJ)=SUR(IJ)
      BP(IJ)=BPR(IJ)
10    CONTINUE
C
C--- CHECK WALL SOUTH BOUNDARY
C
      DO 600 I=2,NIM
      IJ=IMNJ(I)+2
      IF(ITBS(I).EQ.4) THEN
      LB=IJ
      LW=IJ-1
      TEPR=SQRT(TE(IJ)+TET(IJ))
      DELN=DNS(I)
      DXB=XX(IJ-1)-XX(IJ-NJ-1)
      DYB=YY(IJ-1)-YY(IJ-NJ-1)
      RB=HAF*(R(IJ-1)+R(IJ-NJ-1))
      DENS=DEN(IJ)
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
      SUVS(I)=SVU
      SPVS(I)=SVP
      SUWS(I)=SWU
      SPWS(I)=SWP
      GENTS(I)=GENTE
      AS(IJ)=0.0
      ENDIF
C
C-- CHECK NORTH WALL-BOUNDARY
C
      IJ=IMNJ(I)+NJM
      IF(ITBN(I).EQ.4) THEN
      LB=IJ
      LW=IJ+1
      TEPR=SQRT(TE(IJ)+TET(IJ))
      DELN=DNN(I)
      DXB=XX(IJ)-XX(IJ-NJ)
      DYB=YY(IJ)-YY(IJ-NJ)
      RB=HAF*(R(IJ)+R(IJ-NJ))
      DENS=DEN(IJ)
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
      SUVN(I)=SVU
      SPVN(I)=SVP
      SUWN(I)=SWU
      SPWN(I)=SWP
      GENTN(I)=GENTE
      AN(IJ)=0.0
      ENDIF
C
600   CONTINUE
C
C-- CHECK WALL WEST-BOUNDARY
C
      DO 620 J=2,NJM
      IJ=IMNJ(2)+J
      IF(JTBW(J).EQ.4) THEN
      LB=IJ
      LW=IJ-NJ
      TEPR=SQRT(TE(IJ)+TET(IJ))
      DELN=DNW(J)
      DXB=XX(IJ-NJ)-XX(IJ-NJ-1)
      DYB=YY(IJ-NJ)-YY(IJ-NJ-1)
      RB=HAF*(R(IJ-NJ)+R(IJ-NJ-1))
      DENS=DEN(IJ)
```



```
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
      SUVW(J)=SVU
      SPVW(J)=SVP
      SPWW(J)=SWP
      SUWW(J)=SWU
      GENTW(J)=GENTE
      AW(IJ)=0.0
      ENDIF
C
C-- CHECK WALL EAST-BOUNDARY
C
      IJ=IMNJ(NIM)+J
      IF(JTBE(J).EQ.4) THEN
      LB=IJ
      LW=IJ+NJ
      TEPR=SQRT(TE(IJ)+TET(IJ))
      DELN=DNE(J)
      DXB=XX(IJ)-XX(IJ-1)
      DYB=YY(IJ)-YY(IJ-1)
      RB=HAF*(R(IJ)+R(IJ-1))
      DENS=DEN(IJ)
      CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
      SU(IJ)=SU(IJ)+SUU
      BP(IJ)=BP(IJ)+SUP
      SUVE(J)=SVU
      SPVE(J)=SVP
      SPWE(J)=SWP
      SUWE(J)=SWU
      GENTEE(J)=GENTE
      AE(IJ)=0.0
      ENDIF
C
620   CONTINUE
C
      RETURN
      END
C
C
C
C
C--- SUBROUTINE 'WALLFN' TO SET WALL FUNCTIONS
C
      SUBROUTINE WALLFN (LB,LW,VISC,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
     & TAU,SUU,SUP,SVU,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
C
      INCLUDE 'mskemod.h'
C
      UP=U(LB)
      VP=V(LB)
      WP=W(LB)
      UWALL=U(LW)
      VWALL=V(LW)
      WWALL=W(LW)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      CONST=DENS*CMU25*TEPR
      YPLS=DELN*CONST/VISC
      TCOEF=VISC/DELN
      IF(LAY2) GOTO 10
      IF (YPLS.LE.11.63) GO TO 10
      UPLUS=LOG(ELOG*YPLS)/CAPPA
      TCOEF=CONST/UPLUS
   10 CONTINUE
      VPINT=UP*DXB+VP*DYB
      VPX=VPINT*DXB
      VPY=VPINT*DYB
```

```
      TAU=-TCOEF*(VPX*DXB+VPY*DYB)
      ARW=ARW*RB
      SUP=TCOEF*ARW*DXB**2
      SVP=TCOEF*ARW*DYB**2
      SWP=TCOEF*ARW
      CON=TCOEF*ARW*DXB*DYB
      SUU=-CON*VP+UWALL*TCOEF*ARW*DXB**2
      SVU=-CON*UP+VWALL*TCOEF*ARW*DYB**2
      SWU= TCOEF*ARW*WWALL*RB
C
C  FOR MOVING WALL
C
      VPINT=VPINT+WP
      VPINT=ABS(VPINT-SQRT(UWALL*UWALL+VWALL*VWALL+WWALL*WWALL))
C
      GENTE=TCOEF*CONST*ABS(VPINT)/(CAPPA*DENS*DELN)
C
      RETURN
      END
C
C-------------------------------------------------------------
C
C-- mskemod.h
C
C  COMMON BLOCK
      PARAMETER (NX=75)
      PARAMETER (NY=50)
      PARAMETER (NXNY=NX*NY)
      COMMON/KEA1/URFKP,URFKT,URFED,URFEDT,URFVIS,NI,NJ,NIM,
     &  NJM,NINJ,IMNJ(NX),IDIR,ITER,ITBS(NX),ITBN(NX),JTBW(NY),
     &  JTBE(NY),TEST,AKSI,G
      COMMON/KEA2/XX(NXNY),YY(NXNY),FX(NXNY),FY(NXNY),ARE(NXNY),
     &  VOL(NXNY),DNS(NX),DNN(NX),DNW(NY),DNE(NY),R(NXNY)
      COMMON/KEA3/AE(NXNY),AW(NXNY),AN(NXNY),AS(NXNY),AP(NXNY),
     &  SU(NXNY)
      COMMON/KEA4/APU(NXNY),APV(NXNY),BE(NXNY),BW(NXNY),BN(NXNY),
     &  BS(NXNY),BP(NXNY),RES(NXNY),F1(NXNY),F2(NXNY)
      COMMON/KEA5/U(NXNY),V(NXNY),W(NXNY),P(NXNY),GEN(NXNY),
     &  TE(NXNY),TET(NXNY),ED(NXNY),EDT(NXNY),DEN(NXNY),VIS(NXNY),
     &  RW(NXNY)
      COMMON/KEA6/RESOR(11),PRTKP,PRTKT,PRTED,PRTEDT,
     &  SORMAX,VISCOS,ALFA,GREAT,SMALL,CMU,SNSW(NY),
     &  UNW(NY),VNW(NY),PHINW(NY),HAF,QTR,DENSIT,TENOM
      COMMON/KEAA/NSWP(4),SOR(4)
      COMMON/KETURB/ VIS2(NXNY),J2LS,J2LN,I2LE,I2LW,LAY2,
     &                CP1,CP2,CP3,CT1,CT2,CT3
      COMMON/KEUVMOD/ SUVS(NX),SPVS(NX),SUVN(NX),SPVN(NX),
     &               SUVW(NY),SPVW(NY),SUVE(NY),SPVE(NY),
     &               SPWN(NX),SPWS(NX),SPWE(NY),SPWW(NY),
     &               SUWN(NX),SUWS(NX),SUWE(NY),SUWW(NY)
      COMMON /KEGENER/ GENTS(NX),GENTN(NX),GENTW(NY),GENTEE(NY)
      LOGICAL TEST,AKSI,LAY2
```

CHAPTER 4

2D/Axisymmetric Algebraic Stress Turbulence Model

Table of Contents

4.1 Introduction

In this section a description is given of the two-dimensional/Axisymmetric Algebraic Stress turbulence Model (ASM) based on the work of Rodi [1]. The model is coded as a self contained computer program to compute turbulent flow quantities when interfaced with a CFD solver. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in Appendix C.

The module uses as input the mean flow properties, as computed by conventional CFD solvers, and calculates the Reynolds stresses, turbulent kinetic energy and the energy dissipation. It is structured to be self-contained and compatible with many CFD codes. It has been tested as a separate unit at Rocketdyne using the finite-volume REACT code [2]. The module has also been tested independently at the University of Alabama at Huntsville (UAH) using own code MAST.

The module computes turbulent flow quantities in two-dimensional planar or axisymmetric geometry with or without swirl. The standard wall functions and the two-layer model of Chen and Patel [3] are used for the near wall treatment.

4.2 Theory and Model Equations

The Algebraic Stress (ASM) module is based on the work of Rodi [1]. The idea is to simplify or truncate the Reynolds stress equation by approximating the convective and diffusive transport of the Reynolds stresses $\overline{u_i u_j}$ in terms of the corresponding transport of turbulent energy. This allows the transport equation for the stresses to be expressed as a set of algebraic formulae containing the turbulence energy and its rate of dissipation as unknowns in the form:

$$\overline{u_i u_j} = \frac{k}{(P-\varepsilon)} \left[P_{ij} - \frac{2}{3} \delta_{ij} \varepsilon + \Phi_{ij} \right]$$

where P_{ij} = Production and $P = \frac{1}{2} P_{kk}$ and

Φ_{ij} = pressure-strain redistribution

$$\Phi_{ij} = \Phi_{ij,1} + \Phi_{ij,2} + \Phi_{ij,1w} + \Phi_{ij,2w}$$

Rotta's linear return-to-isotropy concept for the non-linear part of

$$\Phi_{ij,1} = -C_1 \frac{\varepsilon}{k} (\overline{u_i u_j} - \frac{2}{3} k \, \delta_{ij})$$

is used and the "isotropization of production" concept for the linear "rapid" part of

$$\Phi_{ij,2} = -C_2 \, (P_{ij} - \frac{2}{3} P \, \delta_{ij})$$

is used. Gibson and Launder [4] concept for the wall reflection terms is used as

$$\Phi_{ij,1w} = C_{1w} \, \rho \frac{\varepsilon}{k} \, (\overline{u_k u_m} \; n_k n_m \delta_{ij} - \frac{3}{2} \overline{u_k u_i} \; n_k n_j - \frac{3}{2} \overline{u_k u_j} \; n_k n_i \,) \, f$$

$$\Phi_{ij,2w} = C_{2w} \, (\Phi_{km,2} \; n_k n_m \delta_{ij} - \frac{3}{2} \Phi_{ik,2} \; n_k n_j - \frac{3}{2} \Phi_{jk,2} \; n_k n_i \,) \, f$$

where (n_i) is the wall-normal unit vector in the i -direction. The wall-distance function (f) represents the ratio of the turbulence length scale $(L_\varepsilon = \frac{k^{3/2}}{\varepsilon})$ and the wall distance and is given as

$$f = (\frac{C_m^{0.75} \; k^{1.5}}{K \; \varepsilon}) \frac{1}{\Delta n}$$

with Δn being the wall-normal distance.

The set of algebraic stress equations can be arranged in the form

$$A_{ij} \; \overline{u^2} + B_{ij} \; \overline{v^2} + C_{ij} \, \overline{w^2} + D_{ij} \, \overline{u \, v} + E_{ij} \; \overline{vw} + F_{ij} \; \overline{uw} = G_{ij}$$

where A_{ij}, B_{ij}, C_{ij}, D_{ij}, E_{ij}, F_{ij}, and G_{ij} are functions of the mean and turbulent flow variables.

The above equation can be solved iteratively in the main flow solver. However, the algebraic system of equations is stiff and convergence difficulties are encountered when solved iteratively. Therefore, the set of equations was cast in the general matrix form $\underline{\mathbf{A}} \; \underline{\mathbf{T}} = \underline{\mathbf{B}}$, where

$$
\underline{A} = \begin{bmatrix}
\frac{3\varepsilon}{2\lambda\kappa} + 2\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & -\frac{V}{r} & 2\frac{\partial U}{\partial y} - \frac{\partial V}{\partial x} & -\frac{\partial W}{\partial y} + \frac{W}{r} & -\frac{\partial W}{\partial x} \\
-\frac{\partial U}{\partial x} & \frac{3\varepsilon}{2\lambda\kappa} + 2\frac{\partial V}{\partial y} & -\frac{V}{r} & 2\frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} & -(\frac{\partial W}{\partial y} + 2\frac{W}{r}) & -\frac{\partial W}{\partial x} \\
-\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & \frac{3\varepsilon}{2\lambda\kappa} + 2\frac{V}{r} & -(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) & 2\frac{\partial W}{\partial y} + \frac{W}{r} & 2\frac{\partial W}{\partial x} \\
\frac{\partial V}{\partial x} & \frac{\partial U}{\partial y} & 0 & \frac{\varepsilon}{\lambda\kappa} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} & 0 & -\frac{W}{r} \\
0 & \frac{\partial W}{\partial y} & -\frac{W}{r} & \frac{\partial W}{\partial x} & \frac{\varepsilon}{\lambda\kappa} + \frac{\partial V}{\partial y} + \frac{V}{r} & \frac{\partial V}{\partial x} \\
\frac{\partial W}{\partial x} & 0 & 0 & \frac{\partial W}{\partial y} & \frac{\partial U}{\partial y} & \frac{\varepsilon}{\lambda\kappa} + \frac{\partial U}{\partial x} + \frac{V}{r}
\end{bmatrix}
$$

$$
\underline{T} = [\rho\,\overline{uu}, \rho\,\overline{vv}, \rho\,\overline{ww}, \rho\,\overline{uv}, \rho\,\overline{vw}, \rho\,\overline{uw}\,]^T
$$

$$
\underline{B} = \begin{bmatrix}
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}(\Phi_{11,1w} + \Phi_{11,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}(\Phi_{22,1w} + \Phi_{22,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}(\Phi_{33,1w} + \Phi_{33,2w}) \\
\frac{1}{(1-C_2)}(\Phi_{12,1w} + \Phi_{12,2w}) \\
\frac{1}{(1-C_2)}(\Phi_{23,1w} + \Phi_{23,2w}) \\
\frac{1}{(1-C_2)}(\Phi_{13,1w} + \Phi_{13,2w})
\end{bmatrix}
$$

where $\lambda = \dfrac{1-C_2}{C_1 - 1 + \dfrac{P}{\rho\varepsilon}}$

The matrix was inverted at each iteration step to obtain a converged solution. The wall function and a two-layer model were built in the module to model the near-wall region.

4.3 Module Evaluation

The ASM module was evaluated by comparison with experimental data of Driver and Seegmiller [5] for the backward facing step and the data of Roback and Johnson [6]. The effect of the wall reflection term is also studied with both wall function and two-layer near wall models. Figures 1a and 1b show the stream-function contours for a backward facing step flow using the wall function and the two-layer near wall models with reattachement length of 5.59H and 5.83H respectively (H is the step height). Figures 2a and 2b show the stream-function contours for the Roback & Johnson confined swirling jet flow using the wall function near wall model, where (a) includes the wall-reflection term in the pressure-strain redistribution term and (b) without the wall reflection term. Figure 3 shows a comparison of the axial velocity along the centerline with and without wall reflection term. The comparisons of the predicted mean axial velocity, mean tangential velocity, turbulent intensities $\overline{uu}^{1/2}$, $\overline{vv}^{1/2}$, $\overline{ww}^{1/2}$, and the Reynolds stress $\overline{uw}^{1/2}$ using the ASM model as compared with the single and multi-scale k-ε models are presented in figures 4 to 9 respectively. The figures in general show that the ASM model used here when combined with the wall function near wall treatment predicts better comparisons without using the wall reflection terms. This may be explained by the fact that the wall reflection terms -whose purpose is to damp normal turbulent intensity normal to the wall as the wall is reached- are not effective when using wall functions near the wall. Similar conclusions were also obtained by the UAH group when testing the ASM module using their code (MAST). Also, in the ASM model, a set of algebraic equations for the Reynolds stresses are solved and there is no boundary conditions are needed for the stresses. This is not the same in the full Reynolds stress model (RSM) where a set of nonlinear differential equations are solved and boundary conditions for the stresses are required. More on this will be discussed in detail in the next RSM module. Also, more details will be given on the tensorial incorporation of the wall reflection terms since they are tied to the orientation of the wall through the unit normal vectors.

REFERENCES

[1] Rodi, W., 'A New Algebraic Stress Relations for Calculating the Reynolds Stresses' Z. Ang. Math. und Mech., vol. 56, pp. 219-, 1976.

[2] Chon J., Hadid A. and Hamakiotes C. 'REACT-2D Version 2.6 User's Manual' CFD Technology Center, Rocketdyne Division/Rockwell International, 1989.

[3] Chen H. and Patel V. 'Near-Wall Turbulence Models for Complex Flows Including Separation', AIAA Journal, vol. 26, No. 6, pp. 641-648, 1988.

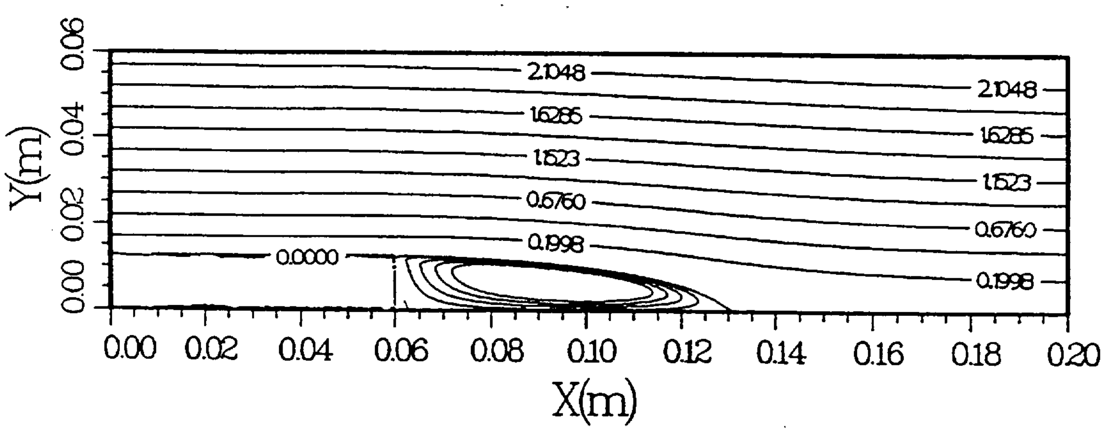
[4] Gibson M. and Launder B., 'Ground Effects on Pressure Fluctuations in the Atmosphere boundary Layer', J. Fluid Mech. vol. 86, pt. 3, pp. 491-., 1978.

[5] Driver, D. and Seegmiller, H. "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow" AIAA J., vol. 23, pp. 163-171, 1985.

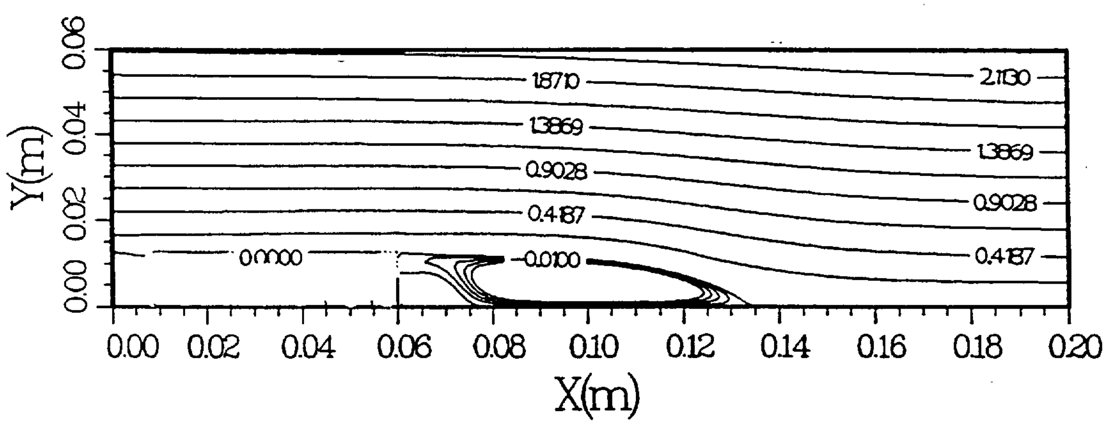
[6] Roback, R. and Johnson, B. "Mass and Momentum Turbulent Transport Experiment with Confined Swirling Co-axial Jets" NASA CR-168252, 1983.

[7] Daly B. and Harlow F., 'Transport Equation in Turbulence', Physics of Fluids, vol. 13, pp. 2634-., 1970.

[8] Stone H., 'Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations', SIAM J. Num. Anal., vol 5, pp. 530- ,1968.

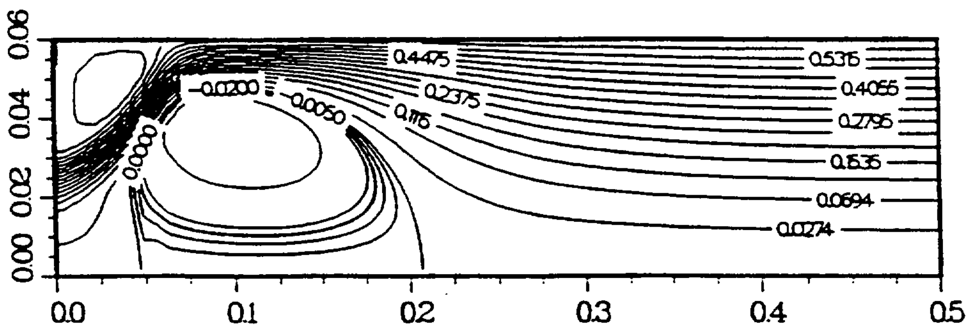


(a)

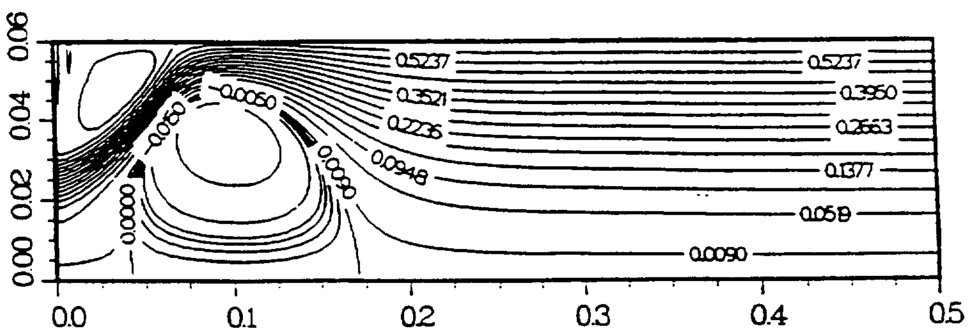


(b)

Figure 1. Stream-function contours of backward facing step flow using the ASM with (a) wall function and (b) two-layer near wall treatment



(a) ASM with Φ_w term



(b) ASM without Φ_w term

Figure 2. Stream-function contours of confined swirling jet flow

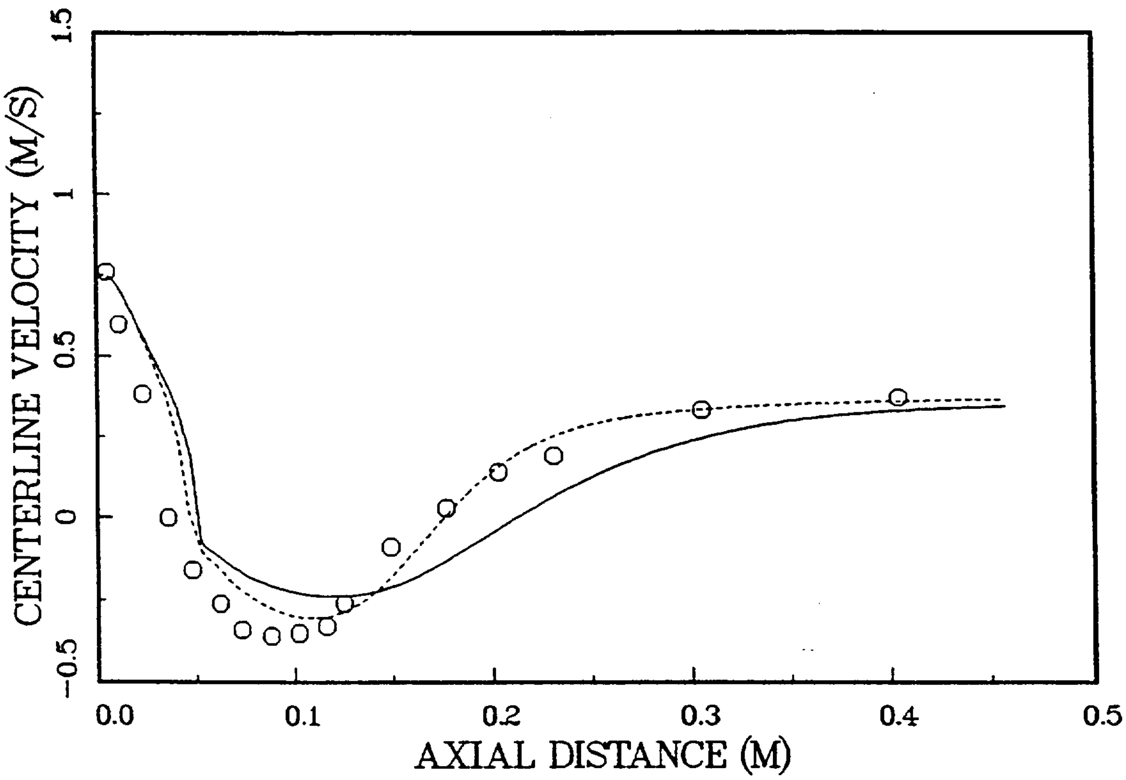


Figure 3. Decay of axial velocity along the centerline in confined swirling jet flow

o	Roback & Johnson
-----	ASM no Φ_w
——	ASm with Φ_w

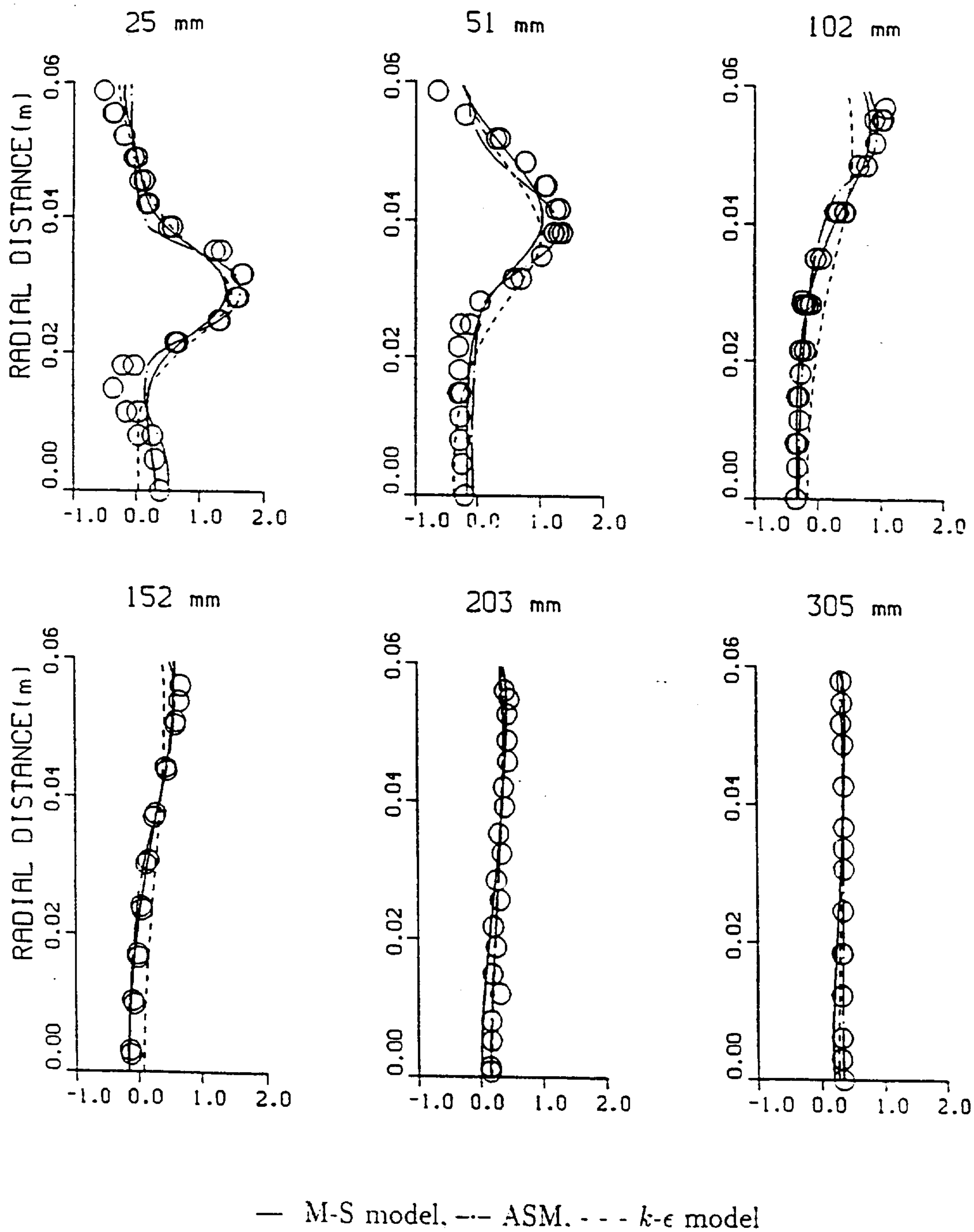
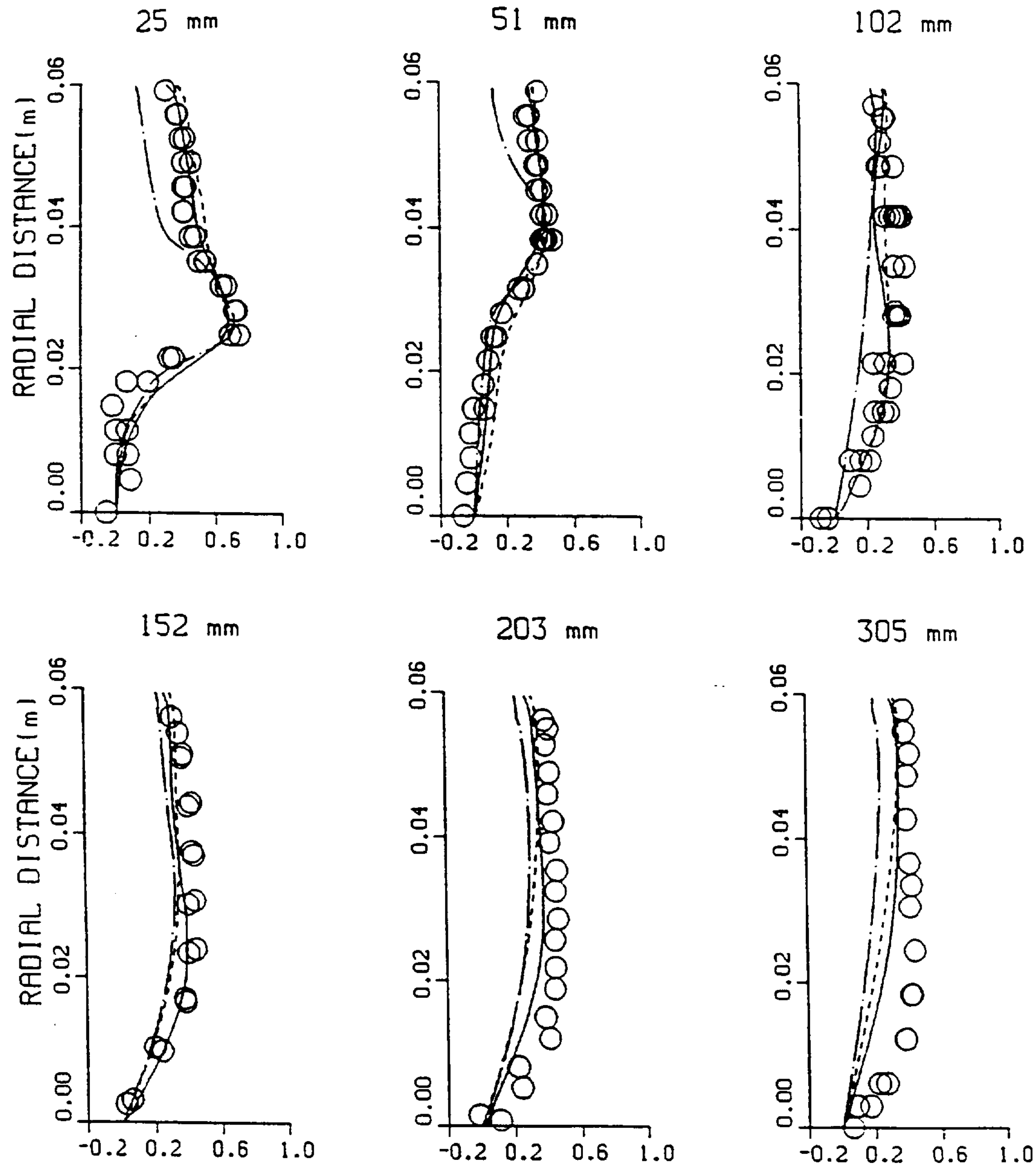
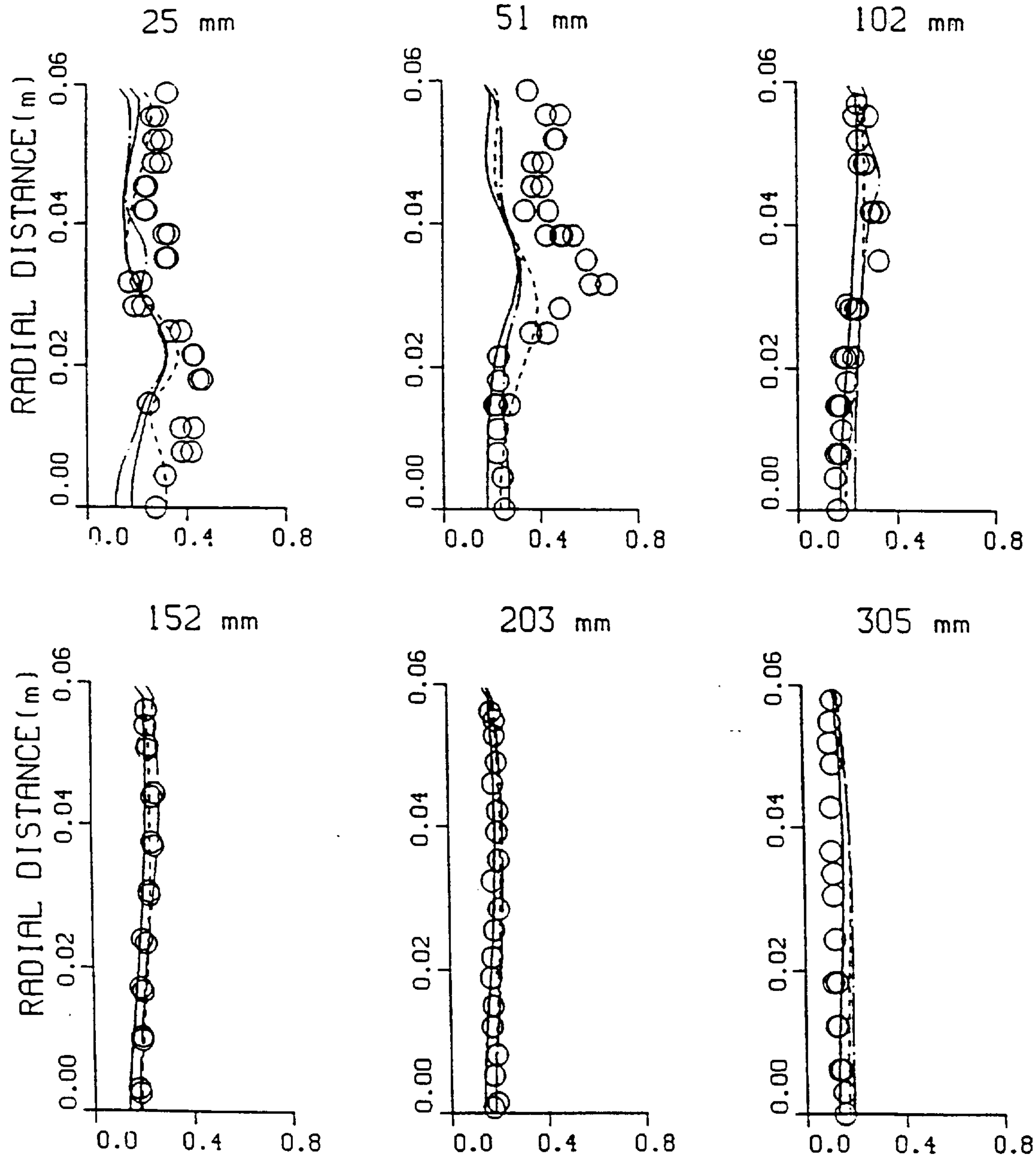


Figure 4. Radial profiles of mean axial velocity in confined swirling jet flow



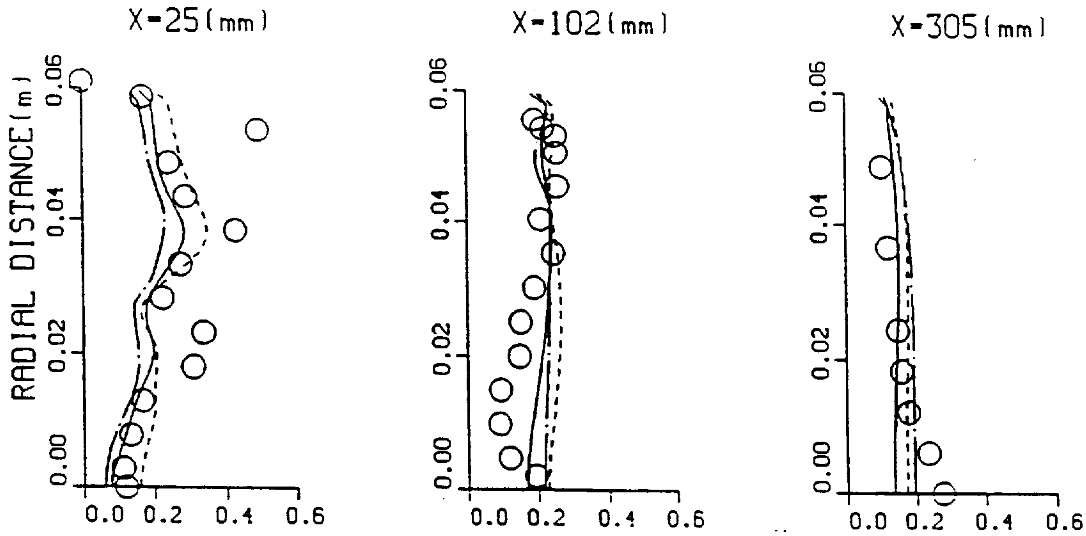
— M-S model. —·— ASM. - - - k-ϵ model

Figure 5. Radial profiles of mean tangential velocity in confined swirling jet flow



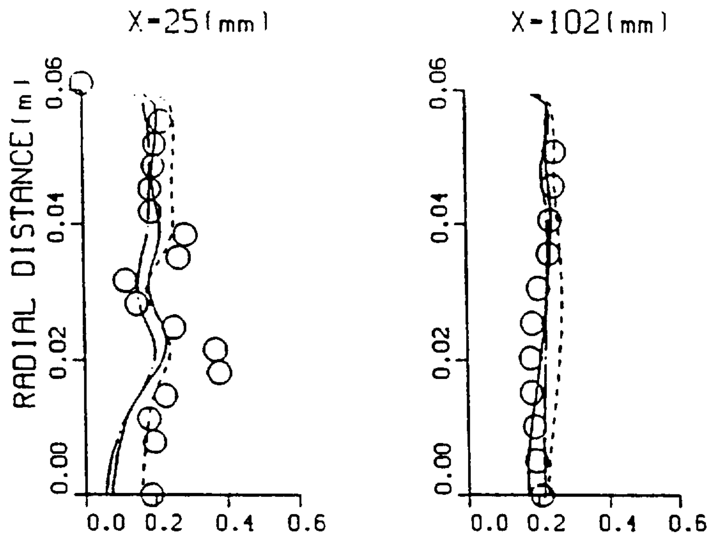
— M-S model. —— ASM. - - - k-ϵ model

Figure 6. Radial profiles of turbulent intensity $(\overline{uu})^{1/2}$ in confined swirling jet flow



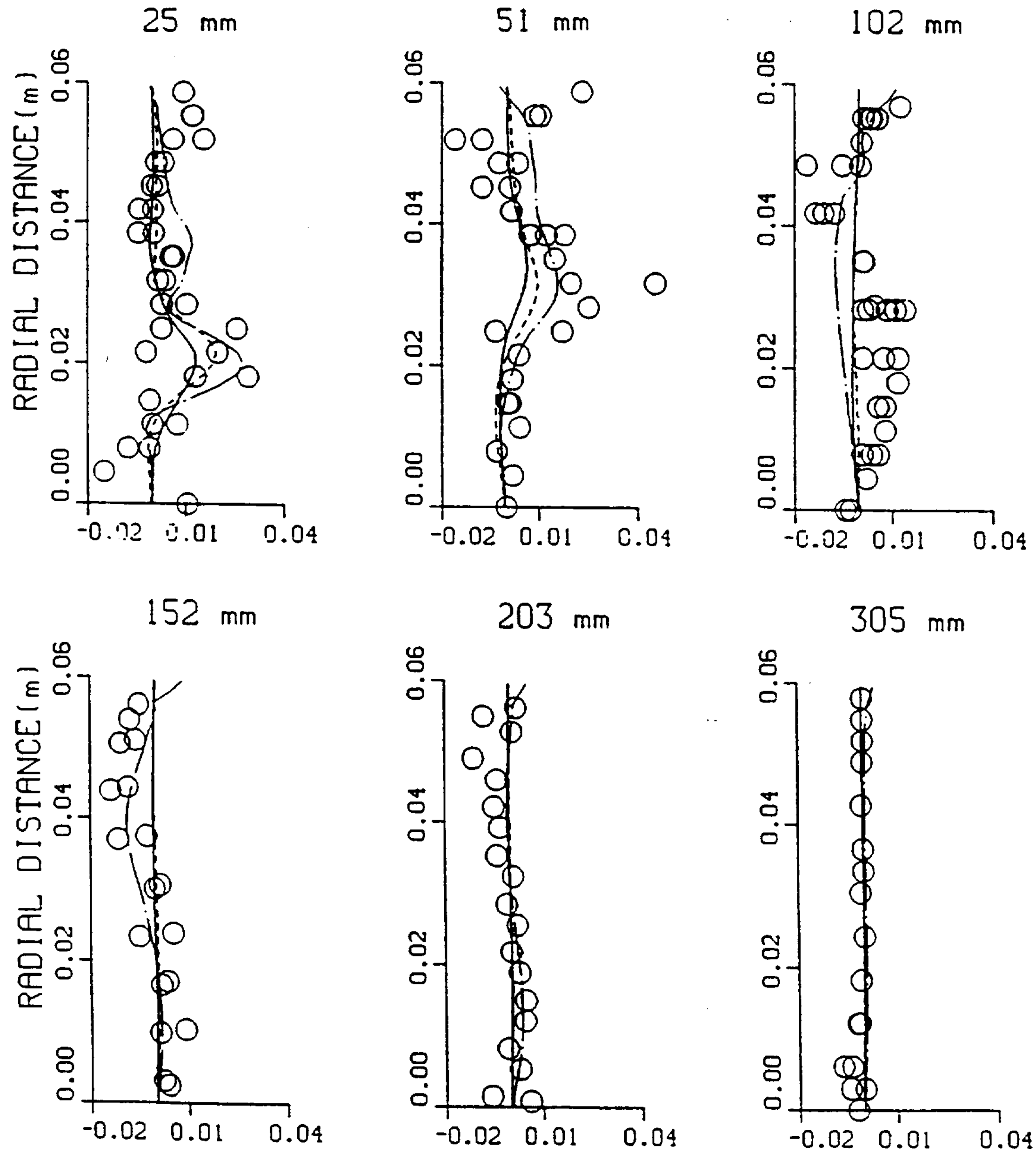
— M-S model, —·— ASM, - - - k-ϵ model

Figure 7. Radial profiles of turbulent intensity $(\overline{vv})^{1/2}$ in confined swirling jet flow



— M-S model, —·— ASM, - - - k-ϵ model

Figure 8. Radial profiles of turbulent intensity $(\overline{ww})^{1/2}$ in confined swirling jet flow



— M-S model, -·- ASM, - - - k-ϵ model

Figure 9. Radial profiles of turbulent shear stress $\overline{uw}$ in confined swirling jet flow

APPENDIX C

2D/Axisymmetric Algebraic Stress Turbulence Module Deck

This module is a FORTRAN source code to solve 2D/Axisymmetric turbulent flow quantities using the algebraic stress model when interfaced with a main flow solver. The module consists of the main routine ASMOD that calls a number of subroutines to perform different functions that will be explained below.

3.1 Subroutine ASMOD

This is basically the main routine that reads through its argument list different variables from the calling flow solver which are described below.

List of Argument Variable Names

X	Grid node locations in the x or ξ-direction, dimensioned to X(NX*NY)
Y	Grid node locations in the y or η-direction, dimensioned to Y(NX*NY)
FX	Interpolation factor in the x or ξ-direction.
FY	Interpolation factor in the y or η-direction.
ARE	Cell areas
VOL	Cell volumes.
R	Radial distance in the axisymmetric geometry or 1. for planar geometry.
DNS	Normal distance of a cell from the south-boundary dimensioned to NX.
DNN	Normal distance of a cell from the north-boundary dimensioned to NX.
DNE	Normal distance of a cell from the east-boundary dimensioned to NY.
DNW	Normal distance of a cell from the west-boundary dimensioned to NY.
U	Axial or x-direction velocity, dimensioned to NX*NY.
V	Radial or y-direction velocity, dimensioned to NX*NY.
W	Tangential or azimuthal velocity, dimensioned to NX*NY.
TE	Turbulent kinetic energy, dimensioned to NX*NY.
ED	Turbulent energy dissipation rate, dimensioned to NX*NY.
DEN	Density (assumed constant for incompressible flows).
F1	Mass flux at cell faces in the x or ξ-direction, dimensioned to NX*NY.
F2	Mass flux at cell faces in the y or η-direction, dimensioned to NX*NY.

VISCOUS	Laminar viscosity.
VIS	Eddy viscosity, dimensioned to NX*NY.
RESOR	Residual error for the k and ε-equations solver, dimensioned to 2.
ITBS	Boundary condition flag along the south boundary dimensioned to NX and must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall, e.g., for a wall boundary condition along the south boundary set ITBS to NX*4. Similarly for the other boundaries.
ITBN	Boundary condition flag along the north boundary, dimensioned to NX.
JTBE	Boundary condition flag along the east-boundary dimensioned to NY.
JTBW	Boundary condition flag along the west-boundary dimensioned to NY.
ITER	Iteration number.
FMUU	function used in the two-layer model.
ICAL	= 1 for swirl velocity calculations, 0 otherwise.
AKSI	= 1 for axisymmetric flow, 0 otherwise.
RESTART	= 1 if calculations are restarted from a previous run, 0 otherwise.

ASMOD starts by reading the turbulent flow constants, under-relaxation factors and Prandtl/Schmidt numbers for the k and ε equations. These are;

CD1, CD2	constants in the k and ε-equations and are usually set to 1.44 and 1.92 respectively.
CMU, ELOG, and CAPPA	constants in the k and ε-equations and are usually set to 0.09, 9.8 and 0.42 respectively.
LAY2	set to true (T) for two-layer model and false (F) for wall functions.
GKE	is set to 1 for second-order upwinding of the convective terms in the k and ε-equations.
ALFAKE	is the iteration parameter used in the k and ε-equation solver.
URFVIS	is the underrelaxation factor of the viscosity near the wall.
SORKE(1) and SORKE (2)	are the degree of accuracy for the k and ε-equation solver respectively.
URFKE(1) and URFKE(2)	are the underrelaxation factors for the k and ε-equations respectively.
PRTKE(1) and PRTKE(2)	are ratio of Prandtl to Schmidt numbers used in the k and ε-equations in the two-layer model near the wall.
C1, C2	are constants in the ASM model.

C1W and C2W	are the two constants in the wall-reflection terms of the pressure-strain redistribution term.
CK and CE	constants in the diffusion term of the k and ε -equations.
WREFON	= 1 if the wall reflection terms of the pressure-strain term are to be included, o otherwise.

All dimensions considered are one-dimensional. The position of any node is defined as $IJ = (I,J) = (I-1)*NJ + J$, where NI and NJ are the number of grid nodes in the X and Y-directions respectively. It is assumed that grid related data such as cell areas, volumes and interpolation factors be passed to the module from an external grid generator.

Subroutine WALREF

This subroutine calculates the wall reflection terms in the pressure-strain redistribution term. It calculates the wall unit normal vectors and the normal distance away from the wall. This is needed to resolve the wall tangential and normal velocity components that are needed to obtain the near-wall values of the Reynolds stresses.

Subroutine CALPIJ

This subroutine calculates the production terms of the individual stress components.

Subroutine CALUIUJ

This subroutine calculates the individual stress component from its algebraic equation. It sets the coefficients of the algebraic stress equations which are solved implicitly at each iteration step by inverting a 6x6 matrix.

Subroutine ACALCKE

This subroutine solves the transport equations for the turbulent energy (IPHI=1) and energy dissipation.(IPHI=2). Daly and Harlow [7] gradient stress diffusion form is used in the module instead of the simplified isotropic diffusivity form. The subroutine calls MODPHI subroutine that sets the appropriate boundary conditions for k and ε. The set of algebraic difference equations are then solved using Stone's strongly implicit solver ASOLSIP.

Subroutine ATWOLAY

This subroutine calculates the near wall turbulence using Chen and Patel's [3] two layer model.

Subroutine MODPIJ

This subroutine modifies the production terms near the wall using the near wall region model.

Subroutine MODPHI

This subroutine calculates the near wall boundary conditions for the turbulence energy and the energy dissipation.

Subroutine AMODVIS

This subroutine modifies the eddy viscosity close to a wall using the near wall model chosen.

Subroutine ASOLSIP

This subroutine solves the system of linear algebraic equations for k and ε using Stone's Implicit Procedure [8].

Subroutine AMODIFY

This subroutine is called from the momentum equations solver of the main routine. It updates the flux source terms of the discretized momentum equations due to wall shear stresses and due to the Reynolds stress gradients. The terms SUASM, SVASM and SWASM need to be added to the U, V and W-momentum equations of the main solver. They represent the difference form of the Reynold stress gradients in the momentum equations.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                   C
C     2D/AXISYMMETRIC ALGEBRAIC STRESS TURBULENCE MODULE            C
C                                                                   C
C     Rocketdyne CFD Technology Center                              C
C                                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C-----------------------------------------------------------------
      SUBROUTINE ASMOD (X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,
     &                  U,V,W,TE,ED,DEN,F1,F2,VISCOS,VIS,
     &                  RESOR,ITBS,ITBN,JTBE,JTBW,ITER,
     &                  FMUU,ICAL,AKSI,RESTART)
C------------------------------------------------------------------
C
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
C
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),
     &          ARE(NXNY),VOL(NXNY),R(NXNY)
      DIMENSION DNS(NX),DNN(NX),DNE(NY),DNW(NY)
      DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)
      DIMENSION TE(NXNY),ED(NXNY),VIS(NXNY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY),
     &          F1(NXNY),F2(NXNY),FMUU(NXNY)
      DIMENSION PHI(NXNY),RESOR(2),TERS(NXNY)
C
C
      NI=NIM+1
      NJ=NJM+1
      NINJ=NI*NJ
C
      IF(ITER.EQ.1) THEN
C
      REWIND 41
C
      READ(41,*)
      READ(41,*)
      READ(41,*) CD1,CD2,CMU,ELOG,CAPPA
      READ(41,*)
      READ(41,*) LRE,LAY2
      READ(41,*)
      READ(41,*) GKE,ALFAKE,URFVIS
      READ(41,*)
      READ(41,*) SORKE(1),NSWPKE(1),URFKE(1),PRTKE(1)
      READ(41,*)
      READ(41,*) SORKE(2),NSWPKE(2),URFKE(2),PRTKE(2)
      READ(41,*)
      READ(41,*) C1,C2,C1W,C2W,CK,CE
      READ(41,*)
      READ(41,*) WREFON
      DO 15 IJ=1,NINJ
      FMU(IJ)=1.
15    CONTINUE
      END IF
C
C
C--- CALCULATE WALL REFLECTION TERMS
C
      IF (WREFON.EQ.1.) CALL WALREF(X,Y,TE,ED,ITBS,ITBN,JTBW,JTBE)
C
C--- CALCULATE TURBULENCE PRODUCTION TERMS
C
      CALL CALPIJ (X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,U,V,W,
     &             ITBS,ITBN,JTBE,JTBW,iter)
C
C--- CALCULATE REYNOLDS STRESSES FROM ALGEBRAIC EQUATIONS
C
      CALL CALUIUJ(ICAL,AKSI,R,U,V,W,DEN,TE,ED,VIS,VISCOS,
     &             ITBS,ITBN,JTBE,JTBW,WREFON,iter)
C
```



```
C--- CALCULATE TURBULENT KINETIC ENERGY.
C
      IPHI=1
      DO 20 IJ=1,NINJ
20    PHI(IJ)=TE(IJ)
C
      CALL ACALCKE(PHI,X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,
     &             U,V,W,DEN,F1,F2,VIS,VISCOS,TE,ED,
     &             ITBS,ITBN,JTBE,JTBW,AKSI,IPHI,RESOR)
C
      DO 25 IJ=1,NINJ
25    TE(IJ)=PHI(IJ)
C
      IF (LAY2) CALL ATWOLAY(X,Y,DEN,TE,ED,VISCOS,ITBS,ITBN,JTBW,JTBE)
C
C--- CALCULATE TURBULENT KINETIC ENERGY DISSIPATION
C
      IPHI=2
      DO 30 IJ=1,NINJ
30    PHI(IJ)=ED(IJ)
C
      CALL ACALCKE(PHI,X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,
     &             U,V,W,DEN,F1,F2,VIS,VISCOS,TE,ED,
     &             ITBS,ITBN,JTBE,JTBW,AKSI,IPHI,RESOR)
C
      DO 35 IJ=1,NINJ
35    ED(IJ)=PHI(IJ)
C
      CALL AMODVIS (TE,ED,VIS,DEN,VISCOS,
     1                 ITBS,TITBN,JTBE,JTBW)
C--- CHECK TURBULENT ENERGY CALCULATED (TE) WITH BELOW (TERS)
C
      DO 40 IJ=1,NINJ
      FMUU(IJ)=FMU(IJ)
      TERS(IJ)=0.5*(U2(IJ)+V2(IJ)+W2(IJ))
40    CONTINUE
      if(mod(iter,10).eq.1) write(*,7) TERS(923)/DEN(923),TE(923)
7     format(1x,7(1x,E10.3))
C
      RETURN
      END
C
C-----------------------------------------------------
      SUBROUTINE WALREF (X,Y,TE,ED,ITBS,ITBN,JTBW,JTBE)
C-----------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
      DIMENSION X(NXNY),Y(NXNY),TE(NXNY),ED(NXNY)
      DIMENSION FN1S(NX),FN1N(NX),FN1E(NY),FN1W(NY)
      DIMENSION FN2S(NX),FN2N(NX),FN2E(NY),FN2W(NY)
      DIMENSION FT1S(NX),FT1N(NX),FT1E(NY),FT1W(NY)
      DIMENSION FT2S(NX),FT2N(NX),FT2E(NY),FT2W(NY)
      DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
C
C--- CALCULATE NORMAL AND TANGENTIAL WALL UNIT VECTORS
C
      NI=NIM+1
      NJ=NJM+1
C--- ALONG SOUTH & NORTH WALLS---
C
      DO 10 I=2,NIM
      IF(ITBS(I).EQ.4) THEN
      IJ=IMNJ(I)+1
      DXB=X(IJ)-X(IJ-NJ)
      DYB=Y(IJ)-Y(IJ-NJ)
      FHIP=SQRT(DXB**2+DYB**2)
      FT1S(I)=DXB/FHIP
      FT2S(I)=DYB/FHIP
      FN1S(I)=-DXB/FHIP
      FN2S(I)=DXB/FHIP
      ENDIF
```

```
      IF(ITBN(I).EQ.4) THEN
      IJ=IMNJ(I)+NJ
      DXB=X(IJ)-X(IJ-NJ)
      DYB=Y(IJ)-Y(IJ-NJ)
      FHIP=SQRT(DXB**2+DYB**2)
      FT1N(I)=DXB/FHIP
      FT2N(I)=DYB/FHIP
      FN1N(I)=-DYB/FHIP
      FN2N(I)=DXB/FHIP
      ENDIF
 10   CONTINUE
C
C--- ALONG WEST & EAST BOUNDARIES ---
C
      DO 20 J=2,NJM
      IF(JTBW(J).EQ.4) THEN
      IJ=J
      DXB=X(IJ)-X(IJ-1)
      DYB=Y(IJ)-Y(IJ-1)
      FHIP=SQRT(DXB*82+DYB**2)
      FT1W(J)=DXB/FHIP
      FT2W(J)=DYB/FHIP
      FN1W(J)=-DYB/FHIP
      FN2W(J)=DXB/FHIP
      ENDIF
      IF(JTBE(J).EQ.4) THEN
      IJ=IMNJ(NIM)+J
      DXB=X(IJ)-X(IJ-1)
      DYB=Y(IJ)-Y(IJ-1)
      FHIP=SQRT(DXB**2+DYB**2)
      FT1E(J)=DXB/FHIP
      FT2E(J)=DYB/FHIP
      FN1E(J)=-DYB/FHIP
      FN2E(J)=DXB/FHIP
      ENDIF
 20   CONTINUE
C
      CMU25=SQRT(SQRT(CMU))
      CMU75=CMU25**3
      FCON=CMU75/CAPPA
C
      DO 80 I=2,NIM
      DO 80 J=2,NJM
      IJ=IMNJ(I)+J
      RDISNS=0.0
      RDISNN=0.0
      RDISNE=0.0
      RDISNW=0.0
      TE(IJ)=ABS(TE(IJ))
      COEF=FCON*TE(IJ)**1.5/(ED(IJ)+SMALL)
C---START WITH SOUTH BOUNDARY
      IF(ITBS(I).EQ.4) THEN
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=X(IJW)-X(IMJW)
      DYB=Y(IJW)-Y(IMJW)
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNS=1.0/(DISNS+SMALL)
      ENDIF
C---CHECK NORTH BOUNDARY
      IF(ITBN(I).EQ.4) THEN
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=X(IJW)-X(IMJW)
      DYB=Y(IJW)-Y(IMJW)
```



```
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNN=1.0/(DISNN+SMALL)
      ENDIF
C---ALONG THE WEST BOUNDARY
      IF(JTBW(J).EQ.4) THEN
      IJW=J
      IMJW=IJW-1
      DXB=X(IJW)-X(IMJW)
      DYB=Y(IJW)-Y(IMJW)
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNW=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNW=1.0/(DISNW+SMALL)
      ENDIF
C---CHECK EAST BOUNDARY
      IF(JTBE(J).EQ.4) THEN
      IJW=IMNJ(NIM)+J
      IMJW=IJW-1
      DXB=X(IJW)-X(IMJW)
      DYB=Y(IJW)-Y(IMJW)
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNE=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNE=1.0/(DISNE+SMALL)
      ENDIF
C
      FUNX(IJ)=COEF*(RDISNS*FN1S(I)**2+RDISNN*FN1N(I)**2+
     &              RDISNE*FN1E(J)**2+RDISNW*FN1W(J)**2)
      FUNY(IJ)=COEF*(RDISNS*FN2S(I)**2+RDISNN*FN2N(I)**2+
     &              RDISNE*FN2E(J)**2+RDISNW*FN2W(J)**2)
      FUNXY(IJ)=COEF*(RDISNS*FN1S(I)*FN2S(I)+
     &  RDISNN*FN1N(I)*FN2N(I)+RDISNE*FN1E(J)*FN2E(J)+
     &  RDISNW*FN1W(J)*FN2W(J))
C
C
      RETURN
      END
C
C
C-----------------------------------------------------------
      SUBROUTINE CALPIJ (X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,U,V,W,
     &                  ITBS,ITBN,JTBE,JTBW,iter)
C-----------------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
c
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY)
      DIMENSION ARE(NXNY),VOL(NXNY),R(NXNY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY)
      DIMENSION U1W(NY),U2W(NY),U3W(NY)
      DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)
c
C
      NI=NIM+1
      NJ=NJM+1
      DO 10 J=2,NJM
      IJ=J
```

```
      IJM=IJ-1
      IPJ=IJ+NJ
      U1W(J)=U(IJ)
      U2W(J)=V(IJ)
      U3W(J)=W(IJ)
 10   CONTINUE
C
      DO 101 I=2,NIM
      J=1
      IJ=IMNJ(I)+J
      UN=U(IJ)
      VN=V(IJ)
      WN=W(IJ)
      DO 102 J=2,NJM
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      IJP=IJ+1
      IJM=IJ-1
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.-FYN
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      DXEW=HAF*(X(IJ)-X(IMJ)+X(IJM)-X(IMJ-1))
      DXNS=HAF*(X(IJ)-X(IJM)+X(IMJ)-X(IMJ-1))
      DYEW=HAF*(Y(IJ)-Y(IMJ)+Y(IJM)-Y(IMJ-1))
      DYNS=HAF*(Y(IJ)-Y(IJM)+Y(IMJ)-Y(IMJ-1))
C
      US=UN
      VS=VN
      WS=WN
C
      UN=U(IJ)*FYS+U(IJP)*FYN
      VN=V(IJ)*FYS+V(IJP)*FYN
      WN=W(IJ)*FYS+W(IJP)*FYN
C
      UE=U(IJ)*FXW+U(IPJ)*FXE
      VE=V(IJ)*FXW+V(IPJ)*FXE
      WE=W(IJ)*FXW+W(IPJ)*FXE
C
      DUEW=UE-U1W(J)
      DUNS=UN-US
      DVEW=VE-U2W(J)
      DVNS=VN-VS
      DWEW=WE-U3W(J)
      DWNS=WN-WS
C
      DUDX(IJ)=(DUEW*DYNS-DUNS*DYEW)/ARE(IJ)
      DUDY(IJ)=(DUNS*DXEW-DUEW*DXNS)/ARE(IJ)
      DVDX(IJ)=(DVEW*DYNS-DVNS*DYEW)/ARE(IJ)
      DVDY(IJ)=(DVNS*DXEW-DVEW*DXNS)/ARE(IJ)
      DWDX(IJ)=(DWEW*DYNS-DWNS*DYEW)/ARE(IJ)
      DWDY(IJ)=(DWNS*DXEW-DWEW*DXNS)/ARE(IJ)
C
C
       if(iter.eq.1) then
       P11(IJ)=-2.*(U2(IJ)*DUDX(IJ)+UV(IJ)*DUDY(IJ))
       P22(IJ)=-2.*(UV(IJ)*DVDX(IJ)+V2(IJ)*DVDY(IJ))
       IF(ICAL.EQ.1) P22(IJ)=P22(IJ)+2.*VW(IJ)*W(IJ)/RP
       P33(IJ)=0.0
       IF(AKSI) P33(IJ)=-2.*W2(IJ)*V(IJ)/RP
       IF(ICAL.EQ.1) P33(IJ)=P33(IJ)-2.*(UW(IJ)*DWDX(IJ)+
     &                                      VW(IJ)*DWDY(IJ))
       P12(IJ)=-(U2(IJ)*DVDX(IJ)+V2(IJ)*DUDY(IJ)+
     #          UV(IJ)*(DUDX(IJ)+DVDY(IJ)))
       IF(ICAL.EQ.1) P12(IJ)=P12(IJ)+UW(IJ)*W(IJ)/RP
       P13(IJ)=0.0
       P23(IJ)=0.0
       IF(ICAL.EQ.1) THEN
       P13(IJ)=-(U2(IJ)*DWDX(IJ)+UV(IJ)*DWDY(IJ)+
```



```
     &           VW(IJ)*DUDY(IJ)+UW(IJ)*DUDX(IJ))
      P23(IJ)=-(UV(IJ)*DWDX(IJ)+UW(IJ)*DVDX(IJ)+
     &           V2(IJ)*DWDY(IJ)+VW(IJ)*DVDY(IJ)-
     &           W2(IJ)*W(IJ)/RP)
      ENDIF
      IF(AKSI) THEN
      P13(IJ)=P13(IJ)-UW(IJ)*V(IJ)/RP
      P23(IJ)=P23(IJ)-VW(IJ)*V(IJ)/RP
      END IF
      GEN(IJ)=0.5*ABS(P11(IJ)+P22(IJ)+P33(IJ))
      end if
C
      U1W(J)=UE
      U2W(J)=VE
      U3W(J)=WE
C
 102  CONTINUE
 101  CONTINUE
      RETURN
      END
C
C--------------------------------------------------------------
      SUBROUTINE CALUIUJ(ICAL,AKSI,R,U,V,W,DEN,TE,ED,VIS,VISCOS,
     &                  ITBS,ITBN,JTBE,JTBW,WREFON,iter)
C--------------------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
      DIMENSION PHI(NXNY),FXWW(NY),DW(NY)
c
      DIMENSION U(NXNY),V(NXNY),W(NXNY),TE(NXNY),ED(NXNY),
     #          DEN(NXNY),R(NXNY),A(6,6),B(6),VIS(NXNY)
      DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)
c
      NI=NIM+1
      NJ=NJM+1
C
C----CALCULATES ALGEBRAIC STRESS EQUATIONS IN THE FORM
C
C       Ai1*U2(IJ)+Ai2*V2(IJ)+Ai3*W2(ij)+Ai4*UV(ij)+Ai5*VW(IJ)+Ai6*UW(IJ)=
Bi
C
      RELT=0.5
      FERTA1=0.
      FERTA2=0.
      DO 10 I=2,NIM
      DO 10 J=2,NJM
      IJ=IMNJ(I)+J
      RP=QTR*(R(IJ)+R(IJ-1)+R(IJ-NJ)+R(IJ-NJ-1))
      EDK=ED(IJ)/(TE(IJ)+SMALL)
      AUX=(1.-C2)/(C1*ED(IJ)+(GEN(IJ)/DEN(IJ)-ED(IJ)))
      AUX=1./(AUX*TE(IJ)+SMALL)
c.....ORDER OF VARABLE U2,V2,W2,UV,VW,UW
      A(1,1)=1.5*AUX+2.*DUDX(IJ)
      A(1,2)=-DVDY(IJ)
      A(1,3)=0.
      IF(AKSI) A(1,3)=A(1,3)-V(IJ)/RP
      A(1,4)=-DVDX(IJ)+2.*DUDY(IJ)
      A(1,5)=-DWDY(IJ)
      IF(ICAL.EQ.1) A(1,5)=A(1,5)+W(IJ)/RP
      A(1,6)=-DWDX(IJ)
C
      A(2,1)=-DUDX(IJ)
      A(2,2)=1.5*AUX+2.*DVDY(IJ)
      A(2,3)=0.
      IF(AKSI) A(2,3)=A(2,3)-V(IJ)/RP
      A(2,4)=-DUDY(IJ)+2.*DVDX(IJ)
      A(2,5)=-DWDY(IJ)
      IF(ICAL.EQ.1) A(2,5)=A(2,5)-2.*W(IJ)/RP
      A(2,6)=-DWDX(IJ)
C
      A(3,1)=-DUDX(IJ)
```

```
      A(3,2)=-DVDY(IJ)
      A(3,3)=1.5*AUX
      IF(AKSI) A(3,3)=A(3,3)+2.*V(IJ)/RP
      A(3,4)=-DUDY(IJ)-DVDX(IJ)
      A(3,5)=2.*DWDY(IJ)
      IF(ICAL.EQ.1) A(3,5)=A(3,5)+W(IJ)/RP
      A(3,6)=2.*DWDX(IJ)
C
      A(4,1)=DVDX(IJ)
      A(4,2)=DUDY(IJ)
      A(4,3)=0.
      A(4,4)=AUX+DUDX(IJ)+DVDY(IJ)
      A(4,5)=0.
      A(4,6)=0.
      IF(ICAL.EQ.1) A(4,6)=A(4,6)-W(IJ)/RP
C
      A(5,1)=0.
      A(5,2)=DWDY(IJ)
      A(5,3)=0.
      IF(ICAL.EQ.1) A(5,3)=A(5,3)-W(IJ)/RP
      A(5,4)=DWDX(IJ)
      A(5,5)=AUX+DVDY(IJ)
      IF(AKSI) A(5,5)=A(5,5)+V(IJ)/RP
      A(5,6)=DVDX(IJ)
C
      A(6,1)=DWDX(IJ)
      A(6,2)=0.
      A(6,3)=0.
      A(6,4)=DWDY(IJ)
      A(6,5)=DUDY(IJ)
      A(6,6)=AUX+DUDX(IJ)
      IF(AKSI) A(6,6)=A(6,6)+V(IJ)/RP
C
      B(1)=AUX*DEN(IJ)*TE(IJ)
      B(2)=B(1)
      B(3)=B(1)
      B(4)=0.
      B(5)=0.
      B(6)=0.
      IF(WREFON.EQ.1.) THEN
      B1=1./(1.-C2)
      B2=C1W*ED(IJ)/(TE(IJ)+SMALL)
      B3=C2*C2W
      Fw1=-2.*U2(IJ)*FUNX(IJ)-UV(IJ)*FUNXY(IJ)+V2(IJ)*FUNY(IJ)
      Fw2=2.*(P11(IJ)-2./3.*GEN(IJ))*FUNX(IJ)+P12(IJ)*FUNXY(IJ)
     #         -(P22(IJ)-2./3.*GEN(IJ))*FUNY(IJ)
      B(1)=B(1)+B1*1.5*(B2*Fw1+B3*Fw2)
      Fw1=U2(IJ)*FUNX(IJ)-UV(IJ)*FUNXY(IJ)-2.*V2(IJ)*FUNY(IJ)
      Fw2=-(P11(IJ)-2./3.*GEN(IJ))*FUNX(IJ)+P12(IJ)*FUNXY(IJ)
     #         +2.*(P22(IJ)-2./3.*GEN(IJ))*FUNY(IJ)
      B(2)=B(2)+1.5*B1*(B2*Fw1+B3*Fw2)
      Fw1=U2(IJ)*FUNX(IJ)+2.*UV(IJ)*FUNXY(IJ)+V2(IJ)*FUNY(IJ)
      Fw2=-(P11(IJ)-2./3.*GEN(IJ))*FUNX(IJ)-2.*P12(IJ)*FUNXY(IJ)
     #         -(P22(IJ)-2./3.*GEN(IJ))*FUNY(IJ)
      B(3)=B(3)+1.5*B1*(B2*Fw1+B3*Fw2)
      Fw1=-(U2(IJ)+V2(IJ))*FUNXY(IJ)-UV(IJ)*(FUNX(IJ)+FUNY(IJ))
      Fw2=(P11(IJ)+P22(IJ)-4./3.*GEN(IJ))*FUNXY(IJ)+
     #               P12(IJ)*(FUNX(IJ)+FUNY(IJ))
      B(4)=B1*(B2*Fw1+B3*Fw2)
      Fw1=-UW(IJ)*FUNXY(IJ)-VW(IJ)*FUNY(IJ)
      Fw2=P13(IJ)*FUNXY(IJ)+P23(IJ)*FUNY(IJ)
      B(5)=B1*(B2*Fw1+B3*Fw2)
      Fw1=-UW(IJ)*FUNX(IJ)-VW(IJ)*FUNXY(IJ)
      Fw2=P13(IJ)*FUNX(IJ)+P23(IJ)*FUNXY(IJ)
      B(6)=B1*(B2*Fw1+B3*Fw2)
      END IF
C
      CALL SOLV(A,B,6)
      U2(IJ)=B(1)*RELT+(1.-RELT)*U2(IJ)
      V2(IJ)=B(2)*RELT+(1.-RELT)*V2(IJ)
      W2(IJ)=B(3)*RELT+(1.-RELT)*W2(IJ)
```



```
      UV(IJ)=B(4)*RELT+(1.-RELT)*UV(IJ)
      VW(IJ)=B(5)*RELT+(1.-RELT)*VW(IJ)
      UW(IJ)=B(6)*RELT+(1.-RELT)*UW(IJ)
C
      TAUMAX=2.*DEN(IJ)*TE(IJ)
      TAUMIN=0.
      U2(IJ)=AMIN1(U2(IJ),TAUMAX)
      U2(IJ)=AMAX1(U2(IJ),TAUMIN)
      V2(IJ)=AMIN1(V2(IJ),TAUMAX)
      V2(IJ)=AMAX1(V2(IJ),TAUMIN)
      W2(IJ)=AMIN1(W2(IJ),TAUMAX)
      W2(IJ)=AMAX1(W2(IJ),TAUMIN)
C
C
      P11(IJ)=-2.*(U2(IJ)*DUDX(IJ)+UV(IJ)*DUDY(IJ))
      P22(IJ)=-2.*(UV(IJ)*DVDX(IJ)+V2(IJ)*DVDY(IJ))
      IF(ICAL.EQ.1) P22(IJ)=P22(IJ)+2.*VW(IJ)*W(IJ)/RP
      P33(IJ)=0.0
      IF(AKSI) P33(IJ)=-2.*W2(IJ)*V(IJ)/RP
      IF(ICAL.EQ.1) P33(IJ)=P33(IJ)-2.*(UW(IJ)*DWDX(IJ)+
     &                                  VW(IJ)*DWDY(IJ))
      P12(IJ)=-(U2(IJ)*DVDX(IJ)+V2(IJ)*DUDY(IJ)+
     #             UV(IJ)*(DUDX(IJ)+DVDY(IJ)))
      IF(ICAL.EQ.1) P12(IJ)=P12(IJ)+UW(IJ)*W(IJ)/RP
      P13(IJ)=0.0
      P23(IJ)=0.0
      IF(ICAL.EQ.1) THEN
      P13(IJ)=-(U2(IJ)*DWDX(IJ)+UV(IJ)*DWDY(IJ)+
     &             VW(IJ)*DUDY(IJ)+UW(IJ)*DUDX(IJ))
      P23(IJ)=-(UV(IJ)*DWDX(IJ)+UW(IJ)*DVDX(IJ)+
     &             V2(IJ)*DWDY(IJ)+VW(IJ)*DVDY(IJ)-
     &             W2(IJ)*W(IJ)/RP)
      ENDIF
      IF(AKSI) THEN
      P13(IJ)=P13(IJ)-UW(IJ)*V(IJ)/RP
      P23(IJ)=P23(IJ)-VW(IJ)*V(IJ)/RP
      END IF
      GEN(IJ)=0.5*ABS(P11(IJ)+P22(IJ)+P33(IJ))
C
 10   CONTINUE
C
C--- MODIFY GEN-TERMS CLOSE TO A WALL
C
      CALL MODPIJ (ITBS,ITBN,JTBW,JTBE)
C
C
      DO I=1,NI
       IJ=IMNJ(I)+1
       U2(IJ)=U2(IJ+1)
       V2(IJ)=V2(IJ+1)
       W2(IJ)=W2(IJ+1)
       UV(IJ)=UV(IJ+1)
       VW(IJ)=VW(IJ+1)
       UW(IJ)=UW(IJ+1)
       IJ=IMNJ(I)+NJ
       U2(IJ)=U2(IJ-1)
       V2(IJ)=V2(IJ-1)
       W2(IJ)=W2(IJ-1)
       UV(IJ)=UV(IJ-1)
       VW(IJ)=VW(IJ-1)
       UW(IJ)=UW(IJ-1)
      END DO
      DO J=2,NJM
       IJ=IMNJ(1)+J
       U2(IJ)=U2(IJ+NJ)
       V2(IJ)=V2(IJ+NJ)
       W2(IJ)=W2(IJ+NJ)
       UV(IJ)=UV(IJ+NJ)
       VW(IJ)=VW(IJ+NJ)
       UW(IJ)=UW(IJ+NJ)
       IJ=IMNJ(NI)+J
```

```
      U2(IJ)=U2(IJ-NJ)
      V2(IJ)=V2(IJ-NJ)
      W2(IJ)=W2(IJ-NJ)
      UV(IJ)=UV(IJ-NJ)
      VW(IJ)=VW(IJ-NJ)
      UW(IJ)=UW(IJ-NJ)
      END DO
       RETURN
       END
C
C
C----------------------------------------------------------
      SUBROUTINE ACALCKE(PHI,X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,
     &           U,V,W,DEN,F1,F2,VIS,VISCOS,TE,ED,
     &           ITBS,ITBN,JTBE,JTBW,AKSI,IPHI,RESOR)
C----------------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
C
      DIMENSION PHI(NXNY),FXWW(NY),DW(NY)
c
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY)
      DIMENSION ARE(NXNY),VOL(NXNY),R(NXNY),VIS(NXNY)
      DIMENSION F1(NXNY),F2(NXNY),TE(NXNY),ED(NXNY)
      DIMENSION DNS(NX),DNN(NX),DNW(NY),DNE(NY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY)
      DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
      DIMENSION PHINW(NY),SNSW(NY),RESOR(2)
c
      NI=NIM+1
      NJ=NJM+1
C
C-----INITIALISATION OF TEMPORALY STORED VARIABLES
C
      URFPHI=1./URFKE(IPHI)
      PRTINVP=1./PRTKE(IPHI)
C
      IJ=1
      PHINE=PHI(IJ)
      PHINW(IJ)=PHINE
      DO 10 J=2,NJM
      IJ=J
      IJM=IJ-1
      IPJ=IJ+NJ
      IJP=IJ+1
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.0-FYN
C
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      DXE=X(IJ)-X(IJM)
      DYE=Y(IJ)-Y(IJM)
cC
      CKK=CK
      IF(IPHI.EQ.2) CKK=CE
      IF(ED(IJ).NE.0.0) THEN
      TERM=HAF*TE(IJ)/ED(IJ)*CKK*(R(IJ)+R(IJM))
      GAMEU2=TERM*U2(IJ)/AREE
      GAMEV2=TERM*V2(IJ)/AREE
      GAMEUV=TERM*UV(IJ)/AREE
      ENDIF
      DW(J)=GAMEU2*DYE**2+GAMEV2*DXE**2
      IF(LAY2.AND.FMU(IJ).LT.0.95) THEN
      VIST=VIS(IJ)-VISCOS
      GAME=HAF*(VISCOS+VIST*PRTINVP)*(R(IJ)+R(IJ-1))
      DW(J)=GAME/AREE*(DXE**2+DYE**2)
      END IF
      PHISE=PHINE
      PHINE=PHI(IJ+1)*FY(IJ)+PHI(IJ)*(1.-FY(IJ))
      PHINW(J)=PHINE
```



```
      SNSW(J)=0.0
      FXWW(J)=1.0
      IF(JTBW(J).EQ.3.OR.JTBW(J).EQ.4) GO TO 10
      DXKS=QTR*(X(IJ+NJ)+X(IJ+NJ-1)-X(IJ)-X(IJ-1))
      DYKS=QTR*(Y(IJ+NJ)+Y(IJ+NJ-1)-Y(IJ)-Y(IJ-1))
      SNSW(J)=-((GAMEU2*DYKS*DYE+GAMEV2*DXKS*DXE)*(PHINE-PHISE)+
     &   2.*GAMEUV*DXE*DYE*(PHI(IPJ)-PHI(IJ))-
     &   GAMEUV*(DXKS*DYE+DYKS*DXE)*(PHINE-PHISE))
      IF(LAY2.AND.FMU(IJ).LT.0.95) THEN
      SNSW(J)=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
      END IF
   10 CONTINUE
C
      DO 100 I=2,NIM
      J=1
      IJ=IMNJ(I)+J
      IMJ=IJ-NJ
      IJP=IJ+1
      FXE=FX(IJ)
      FXW=1.-FX(IJ)
      FYN=FY(IJ)
      FYS=1.0-FYN
      AREN=HAF*(ARE(IJ)+ARE(IJP))
      DXN=X(IJ)-X(IMJ)
      DYN=Y(IJ)-Y(IMJ)
      DXET=QTR*(X(IJP)+X(IJP-NJ)-X(IJ)-X(IMJ))
      DYET=QTR*(Y(IJP)+Y(IJP-NJ)-Y(IJ)-Y(IMJ))
      GAMNU2=0.0
      GAMNV2=0.0
      GAMNUV=0.0
      CKK=CK
      IF(IPHI.EQ.2) CKK=CE
      IF(ED(IJ).NE.0.0) THEN
      TERM=HAF*TE(IJ)/ED(IJ)*CKK*(R(IJ)+R(IMJ))
      GAMNU2=TERM*U2(IJ)/AREN
      GAMNV2=TERM*V2(IJ)/AREN
      GAMNUV=TERM*UV(IJ)/AREN
      ENDIF
      DN=GAMNU2*DYN**2+GAMNV2*DXN**2
      IF(LAY2.AND.FMU(IJ).LT.0.95) THEN
      VIST=VIS(IJ)-VISCOS
      GAMN=HAF*(VISCOS+VIST*PRTINVP)*(R(IJ)+R(IJ-NJ))
      DN=GAMN/AREN*(DXN**2+DYN**2)
      END IF
      FYSS=1.0
      PHINE=PHI(IJ+NJ)*FXE+PHI(IJ)*FXW
      SEWN=0.0
      IF(ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 110
      SEWN=-((GAMNU2*DYN*DYET+GAMNV2*DXN*DXET)*(PHINE-PHINW(J))+
     &  2.*GAMNUV*DXN*DYN*(PHI(IJP)-PHI(IJ))-
     &  GAMNUV*(DXET*DYN+DXN*DYET)*(PHINE-PHINW(J)))
      IF(LAY2.AND.FMU(IJ).LT.0.95) THEN
      SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
      END IF
  110 CONTINUE
      PHINW(J)=PHINE
C
C-----THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES
C
      DO 101 J=2,NJM
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IPJP=IPJ+1
      IMJ=IJ-NJ
      IJP=IJ+1
      IJM=IJ-1
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.-FYN
C
```

```
      DXE=X(IJ)-X(IJM)
      DYE=Y(IJ)-Y(IJM)
      DXN=X(IJ)-X(IMJ)
      DYN=Y(IJ)-Y(IMJ)
      DXKS=QTR*(X(IPJ)-X(IMJ)+X(IPJ-1)-X(IMJ-1))
      DYKS=QTR*(Y(IPJ)-Y(IMJ)+Y(IPJ-1)-Y(IMJ-1))
      DXET=QTR*(X(IJP)-X(IJM)+X(IJP-NJ)-X(IJM-NJ))
      DYET=QTR*(Y(IJP)-Y(IJM)+Y(IJP-NJ)-Y(IJM-NJ))
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      AREN=HAF*(ARE(IJ)+ARE(IJP))
      CKK=CK
      IF(IPHI.EQ.2) CKK=CE
      GAMEU2=HAF*CKK/AREE*(TE(IJ)/(ED(IJ)+SMALL)*U2(IJ)*FXW+
     &   TE(IPJ)/(ED(IPJ)+SMALL)*U2(IPJ)*FXE)*(R(IJ)+R(IJM))
      GAMEV2=HAF*CKK/AREE*(TE(IJ)/(ED(IJ)+SMALL)*V2(IJ)*FXW+
     &   TE(IPJ)/(ED(IPJ)+SMALL)*V2(IPJ)*FXE)*(R(IJ)+R(IJM))
      GAMEUV=HAF*CKK/AREE*(TE(IJ)/(ED(IJ)+SMALL)*UV(IJ)*FXW+
     &   TE(IPJ)/(ED(IPJ)+SMALL)*UV(IPJ)*FXE)*(R(IJ)+R(IJM))
C
      GAMNU2=HAF*CKK/AREN*(TE(IJ)/(ED(IJ)+SMALL)*U2(IJ)*FYS+
     &   TE(IJP)/(ED(IJP)+SMALL)*U2(IJP)*FYN)*(R(IJ)+R(IMJ))
      GAMNV2=HAF*CKK/AREN*(TE(IJ)/(ED(IJ)+SMALL)*V2(IJ)*FYS+
     &   TE(IJP)/(ED(IJP)+SMALL)*V2(IJP)*FYN)*(R(IJ)+R(IMJ))
      GAMNUV=HAF*CKK/AREN*(TE(IJ)/(ED(IJ)+SMALL)*UV(IJ)*FYS+
     &   TE(IJP)/(ED(IJP)+SMALL)*UV(IJP)*FYN)*(R(IJ)+R(IMJ))
C
      IF(LAY2.AND.FMU(IJ).LT.0.95) THEN
      VISE=VIS(IJ)*FXW+VIS(IPJ)*FXE
      VISET=VISE-VISCOS
      GAME=HAF*(VISCOS+VISET*PRTINVP)*(R(IJ)+R(IJM))
      VISN=VIS(IJ)*FYS+VIS(IJP)*FYN
      VISNT=VISN-VISCOS
      GAMN=HAF*(VISCOS+VISNT*PRTINVP)*(R(IJ)+R(IMJ))
      END IF
      DS=DN
      DE=GAMEU2*DYE**2+GAMEV2*DXE**2
      DN=GAMNU2*DYN**2+GAMNV2*DXN**2
      IF(LAY2.AND.FMU(IJ).LT.0.95) THEN
      DE=GAME/AREE*(DXE**2+DYE**2)
      DN=GAMN/AREN*(DXN**2+DYN**2)
      END IF
C
C LINEAR UPWIND DIFFERENCING
C
      AEE=AMIN1(F1(IJ),0.0)*FX(IPJ)*GKE
      AWW=-AMAX1(F1(IMJ),0.0)*(1.0-FXWW(J))*GKE
      AE1=-AMIN1(F1(IMJ),0.0)*FXE*GKE
      AW1=AMAX1(F1(IJ),0.0)*(1.0-FX(IMJ))*GKE
      ANN=AMIN1(F2(IJ),0.0)*FY(IJP)*GKE
      ASS=-AMAX1(F2(IJM),0.0)*(1.0-FYSS)*GKE
      AN1=-AMIN1(F2(IJM),0.0)*FYN*GKE
      AS1=AMAX1(F2(IJ),0.0)*(1.0-FY(IJM))*GKE
C
      AW(IJ)=DW(J)+AMAX1(F1(IMJ),0.0)-AWW
      AE(IJ)=DE-AMIN1(F1(IJ),0.0)-AEE
      AS(IJ)=DS+AMAX1(F2(IJM),0.0)-ASS
      AN(IJ)=DN-AMIN1(F2(IJ),0.0)-ANN
C
      DXKS=QTR*(X(IPJ)-X(IMJ)+X(IPJ-1)-X(IMJ-1))
      DYKS=QTR*(Y(IPJ)-Y(IMJ)+Y(IPJ-1)-Y(IMJ-1))
      DXET=QTR*(X(IJP)-X(IJM)+X(IJP-NJ)-X(IJM-NJ))
      DYET=QTR*(Y(IJP)-Y(IJM)+Y(IJP-NJ)-Y(IJM-NJ))
C
      PHISE=PHINE
      PHINE=(PHI(IJ)*FYS+PHI(IJP)*FYN)*FXW+
     &      (PHI(IPJ)*FYS+PHI(IPJ+1)*FYN)*FXE
      SEWS=SEWN
      SEWN=-((GAMNU2*DYN*DYET+GAMNV2*DXN*DXET)*
     &      (PHINE-PHINW(J))-GAMNUV*(DXET*DYN+DXN*DYET)*
     &      (PHINE-PHINW(J))+2.*GAMNUV*DXN*DYN*
     &      (PHI(IJP)-PHI(IJ)))
```



```
C
      SNSE=-((GAMEU2*DYKS*DYE+GAMEV2*DXKS*DXE)*
     &      (PHINE-PHISE)+2.*GAMEUV*DXE*DYE*
     &      (PHI(IPJ)-PHI(IJ))-GAMEUV*(DXKS*DYE+DYKS*DXE)*
     &      (PHINE-PHISE))
      IF(LAY2.AND.FMU(IJ).LT.0.95) THEN
      SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
      SNSE=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
      END IF
C
      IF(I.EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.
      IF(J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.
C
C LINEAR UPWIND DIFFERENCING
C
      IMJ1=IMJ-NJ
      IMJJ=MAX(1,IMJ1)
      APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS
      APV(IJ)=APV(IJ)+AEE*PHI(IPJ+NJ)+AWW*PHI(IMJJ)+ANN
     &       *PHI(IJP+1)+ASS*PHI(IJM-1)
     &       +AE1*PHI(IPJ)+AW1*PHI(IMJ)+AN1*PHI(IJP)+AS1*PHI(IJM)
      APU(IJ)=AEE+AWW+ANN+ASS+AE1+AW1+AN1+AS1
C
      GO TO (120,130) IPHI
C
C
C-----TURBULENT KINETIC ENERGY SOURCE TERMS
C
  120 CONTINUE
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
      BP(IJ)=APU(IJ)+VOL(IJ)*DEN(IJ)*ED(IJ)/(TE(IJ)+SMALL)
C
      GO TO 150
C
C-----DISSIP.OF TURB. KIN. ENERGY SOURCE TERMS
C
  130 CONTINUE
C
      SU(IJ)=APV(IJ)+VOL(IJ)*CD1*ED(IJ)*ABS(GEN(IJ))/
     &                 (TE(IJ)+SMALL)
      BP(IJ)=APU(IJ)+VOL(IJ)*CD2*DEN(IJ)*ED(IJ)/
     &                 (TE(IJ)+SMALL)
C
  150 CONTINUE
C
      SNSW(J)=SNSE
      PHINW(J)=PHINE
      FYSS=FY(IJM)
      FXWW(J)=FX(IMJ)
      DW(J)=DE
C
  101 CONTINUE
  100 CONTINUE
C
C-----PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS
C
      CALL MODPHI (PHI,IPHI,X,Y,FX,FY,ARE,VOL,R,
     &             DEN,TE,ED,ITBS,ITBN,JTBE,JTBW,
     &             DNS,DNN,DNE,DNW)
C
          IF( IPHI.EQ.2.AND.LAY2 ) THEN
             DO 30 I=2,NIM
                IF (ITBS(I).EQ.4.OR.ITBN(I).EQ.4) THEN
                   DO 310 J=2,NJM
                      IJ=IMNJ(I)+J
                      IF(FMU(IJ).LT.0.95) THEN
                      SU(IJ)=GREAT*ED(IJ)
                      BP(IJ)=GREAT
                      END IF
  310              CONTINUE
                ENDIF
```

```
 30         CONTINUE
C
            DO 40 J=2,NJM
               IF (JTBW(J).EQ.4.OR.JTBE(J).EQ.4) THEN
                  DO 410 I=2,NIM
                     IJ=IMNJ(I)+J
                     IF(FMU(IJ).LT.0.95) THEN
                     SU(IJ)=GREAT*ED(IJ)
                     BP(IJ)=GREAT
                     END IF
 410              CONTINUE
               ENDIF
 40         CONTINUE
         ENDIF
C
C
      DO 200 I=2,NIM
      DO 201 J=2,NJM
      IJ=IMNJ(I)+J
      AP(IJ)=AW(IJ)+AE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)
      AP(IJ)=AP(IJ)*URFPHI
      SU(IJ)=SU(IJ)+(1.-URFKE(IPHI))*AP(IJ)*PHI(IJ)
 201  CONTINUE
 200  CONTINUE
C
C-----SOLVING F.D. EQUATIONS
C
      CALL ASOLSIP (PHI,IPHI,RESOR)
C
      NINJ=NI*NJ
      DO IJ=1,NINJ
      IF(PHI(IJ).LT.0.) PHI(IJ)=ABS(PHI(IJ))
      END DO
      RETURN
      END
C
C
C--------------------------------------------------------
      SUBROUTINE ATWOLAY(X,Y,DEN,TE,ED,VISCOS,ITBS,ITBN,JTBW,JTBE)
C--------------------------------------------------------
C
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
C
C
      DIMENSION X(NXNY), Y(NXNY), DEN(NXNY)
      DIMENSION TE(NXNY), ED(NXNY)
      DIMENSION ITBS(NX), ITBN(NX), JTBE(NY), JTBW(NY)
C
      NI = NIM + 1
      NJ = NJM + 1
C
      CMU25=SQRT(SQRT(CMU))
      CMU75=CMU25**3
      C11=CAPPA/CMU75
      AED=2.0*C11
      AMU=70.0
C
      DO 10 I=2,NIM
C
       DO 110 J=2,NJM
            DISNS=GREAT
            DISNN=GREAT
            DISNW=GREAT
            DISNE=GREAT
          IJ=IMNJ(I)+J
C
C...CHECK THE SOUTH BOUNDARY
C
         IF(ITBS(I).EQ.4) THEN
              IJW=IMNJ(I)+1
```



```
                   IMJW=IJW-NJ
                   DXB=X(IJW)-X(IMJW)
                   DYB=Y(IJW)-Y(IMJW)
                   XPW=HAF*(X(IJW)+X(IMJW))
                   YPW=HAF*(Y(IJW)+Y(IMJW))
                   XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
                   YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
                   DXBP=XBP-XPW
                   DYBP=YBP-YPW
                   DISNS=DELTA(DXB,DYB,DXBP,DYBP)
             END IF
C
C...CHECK THE NORTH BOUNDARY
C
         IF(ITBN(I).EQ.4) THEN
              IJW=IMNJ(I)+NJM
                   IMJW=IJW-NJ
                   DXB=X(IJW)-X(IMJW)
                   DYB=Y(IJW)-Y(IMJW)
                   XPW=HAF*(X(IJW)+X(IMJW))
                   YPW=HAF*(Y(IJW)+Y(IMJW))
                   XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
                   YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
                   DXBP=XBP-XPW
                   DYBP=YBP-YPW
                   DISNN=DELTA(DXB,DYB,DXBP,DYBP)
           END IF
C
C--CHECK WEST BOUNDARY
C
         IF (JTBW(J).EQ.4) THEN
              IJW=J
                   IMJW=IJW-1
                   DXB=X(IJW)-X(IMJW)
                   DYB=Y(IJW)-Y(IMJW)
                   XB=HAF*(X(IJW)+X(IMJW))
                   YB=HAF*(Y(IJW)+Y(IMJW))
                   XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
                   YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
                   DXBP=XBP-XB
                   DYBP=YBP-YB
                   DISNW=DELTA(DXB,DYB,DXBP,DYBP)
           ENDIF
C--CHECK EAST BOUNDARY
         IF (JTBE(J).EQ.4) THEN
              IJW=IMNJ(NIM)+J
                   IMJW=IJW-1
                   DXB=X(IJW)-X(IMJW)
                   DYB=Y(IJW)-Y(IMJW)
                   XB=HAF*(X(IJW)+X(IMJW))
                   YB=HAF*(Y(IJW)+Y(IMJW))
                   XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
                   YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
                   DXBP=XBP-XB
                   DYBP=YBP-YB
                   DISNE=DELTA(DXB,DYB,DXBP,DYBP)
           ENDIF
C
             DISN=AMIN1(DISNS,DISNN,DISNW,DISNE)
             RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS
             RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))
               ALMU=C11*DISN*(1.0-EXP(-RK/AMU))
               ALED=C11*DISN*(1.0-EXP(-RK/AED))
               FMU(IJ)=ALMU/(ALED+SMALL)
               FMU(IJ)=AMIN1(FMU(IJ),1.)
               IF(FMU(IJ).GE.0.95) FMU(IJ)=1.
                 IF(FMU(IJ).LT.0.95) THEN
                    ED(IJ)=SQRT(TE(IJ))**3/ALED
                    VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU
                 END IF
 110  CONTINUE
```

```
 10     CONTINUE
C
        RETURN
        END
C
C-----------------------------------------------------
       SUBROUTINE MODPIJ (ITBS,ITBN,JTBW,JTBE)
C-----------------------------------------------------
       INCLUDE 'gridparam.h'
       INCLUDE 'asm.h'
       DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
       NI=NIM+1
       NJ=NJM+1
C
C-- SOUTHE WALL
C
       DO 1110 I=2,NIM
       IJ=IMNJ(I)+2
       IF(ITBS(I).EQ.4) THEN
       GEN(IJ)=GENTS(I)
       IF(.NOT.LAY2) GEN(IJ)=GENTS(I)
       ENDIF
C
C---- NORTH WALL
C
       IJ=IMNJ(I)+NJM
       IF(ITBN(I).EQ.4) THEN
       GEN(IJ)=GENTN(I)
       IF(.NOT.LAY2) GEN(IJ)=GENTN(I)
       ENDIF
C
 1110  CONTINUE
C
C--- WEST WALL
C
       DO 1120 J=2,NJM
       IJ=IMNJ(2)+J
       IF(JTBW(J).EQ.4) THEN
       GEN(IJ)=GENTW(J)
       IF(.NOT.LAY2) GEN(IJ)=GENTW(J)
       ENDIF
C
C--- EAST WALL
C
       IJ=IMNJ(NIM)+J
       IF(JTBE(J).EQ.4) THEN
       GEN(IJ)=GENTEE(J)
       IF(.NOT.LAY2) GEN(IJ)=GENTEE(J)
       ENDIF
C
 1120  CONTINUE
C
       RETURN
       END
C
C
C-----------------------------------------------
       SUBROUTINE MODPHI (PHI,IPHI,X,Y,FX,FY,ARE,VOL,R,
      &                   DEN,TE,ED,ITBS,ITBN,JTBE,JTBW,
      &                   DNS,DNN,DNE,DNW)
C-----------------------------------------------
       INCLUDE 'gridparam.h'
       INCLUDE 'asm.h'
c
       DIMENSION PHI(NXNY),DEN(NXNY),TE(NXNY),ED(NXNY)
       DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),ARE(NXNY),
      #          VOL(NXNY),R(NXNY)
       DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
       DIMENSION DNS(NX),DNN(NX),DNW(NY),DNE(NY)
c
C
```



```
       NI=NIM+1
       NJ=NJM+1
       CMU25=SQRT(SQRT(CMU))
       CMU75=CMU25**3
C
       GO TO (800,900) IPHI
C
C-----BOUNDARY CONDITIONS FOR KINETIC TURBULENT ENERGY
C
  800  CONTINUE
C-----SOUTH BOUNDARY
       DO 810 I=2,NIM
       IJ=IMNJ(I)+2
       GO TO (811,812,813,814) ITBS(I)
  811  CONTINUE
       SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)
       BP(IJ)=BP(IJ)+AS(IJ)
       GO TO 815
  812  CONTINUE
       PHI(IJ-1)=TE(IJ)
       GOTO 815
  813  CONTINUE
       IJ=IJ-1
       IPJ=IJ+NJ
       IMJ=IJ-NJ
       FXE1=FX(IJ)
       FXE2=FX(IMJ)
       FXW1=1.-FXE1
       FXW2=1.-FXE2
       DXB=X(IJ)-X(IMJ)
       DYB=Y(IJ)-Y(IMJ)
       DXBP=QTR*(X(IJ+1)-X(IJ)+X(IMJ+1)-X(IMJ))
       DYBP=QTR*(Y(IJ+1)-Y(IJ)+Y(IMJ+1)-Y(IMJ))
       FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
       DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
       PHI(IJ)=TE(IJ+1)-DEL*FAC
       IJ=IJ+1
       GOTO 815
  814  CONTINUE
       IF(.NOT.LAY2) GEN(IJ)=GENTS(I)
       SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  815  CONTINUE
       AS(IJ)=0.0
  810  CONTINUE
C-----NORTH BOUNDARY
       DO 820 I=2,NIM
       IJ=IMNJ(I)+NJM
       GO TO (821,822,823,824) ITBN(I)
  821  CONTINUE
       SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)
       BP(IJ)=BP(IJ)+AN(IJ)
       GO TO 825
  822  CONTINUE
       PHI(IJ+1)=TE(IJ)
       GO TO 825
  823  CONTINUE
       IJ=IJ+1
       IPJ=IJ+NJ
       IMJ=IJ-NJ
       FXE1=FX(IJ)
       FXE2=FX(IMJ)
       FXW1=1.-FXE1
       FXW2=1.-FXE2
       DXB=X(IJ)-X(IMJ)
       DYB=Y(IJ)-Y(IMJ)
       DXBP=QTR*(X(IJ-2)-X(IJ)+X(IMJ-2)-X(IMJ))
       DYBP=QTR*(Y(IJ-2)-Y(IJ)+Y(IMJ-2)-Y(IMJ))
       FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
       DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
       PHI(IJ)=TE(IJ-1)-DEL*FAC
       IJ=IJ-1
```

```
      GO TO 825
  824 CONTINUE
      IF(.NOT.LAY2) GEN(IJ)=GENTN(I)
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  825 CONTINUE
      AN(IJ)=0.0
  820 CONTINUE
C-----WEST BOUNDARY
      DO 830 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (831,832,833,834) JTBW(J)
  831 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*TE(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 835
  832 CONTINUE
      PHI(IJ-NJ)=TE(IJ)
      GO TO 835
  833 CONTINUE
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=Y(IJ)-Y(IJM)
      DXB=X(IJ)-X(IJM)
      DXBP=QTR*(X(IJ+NJ)-X(IJ)+X(IJM+NJ)-X(IJM))
      DYBP=QTR*(Y(IJ+NJ)-Y(IJ)+Y(IJM+NJ)-Y(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2
      PHI(IJ)=TE(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GOTO 835
  834 CONTINUE
      IF(.NOT.LAY2) GEN(IJ)=GENTW(J)
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  835 CONTINUE
      AW(IJ)=0.0
  830 CONTINUE
C-----EAST BOUNDARY
      DO 840 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (841,842,843,844) JTBE(J)
  841 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*TE(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 845
  842 CONTINUE
      PHI(IJ+NJ)=TE(IJ)
      GO TO 845
  843 CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=X(IJ)-X(IJM)
      DYB=Y(IJ)-Y(IJM)
      DXBP=QTR*(X(IJ-NJ-NJ)-X(IJ)+X(IJM-NJ-NJ)-X(IJM))
      DYBP=QTR*(Y(IJ-NJ-NJ)-Y(IJ)+Y(IJM-NJ-NJ)-Y(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2
      PHI(IJ)=TE(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 845
  844 CONTINUE
      IF(.NOT.LAY2) GEN(IJ)=GENTEE(J)
```



```
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
  845 CONTINUE
      AE(IJ)=0.0
  840 CONTINUE
C
C--- OBSTACLE TREATMENT IS IGNORED HERE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR DISSIPATION OF KIN. TURB. ENERGY
C
  900 CONTINUE
C-----SOUTH BOUNDARY
      DO 910 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (911,912,913,914) ITBS(I)
  911 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 915
  912 CONTINUE
      PHI(IJ-1)=ED(IJ)
      GO TO 915
  913 CONTINUE
      IJ=IJ-1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=X(IJ)-X(IMJ)
      DYB=Y(IJ)-Y(IMJ)
      DXBP=QTR*(X(IJ+1)-X(IJ)+X(IMJ+1)-X(IMJ))
      DYBP=QTR*(Y(IJ+1)-Y(IJ)+Y(IMJ+1)-Y(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      PHI(IJ)=ED(IJ+1)-DEL*FAC
      IJ=IJ+1
      GOTO 915
  914 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNS(I))*GREAT
      BP(IJ)=GREAT
  915 CONTINUE
      AS(IJ)=0.0
  910 CONTINUE
C-----NORTH BOUNDARY
      DO 920 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (921,922,923,924) ITBN(I)
  921 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 925
  922 CONTINUE
      PHI(IJ+1)=ED(IJ)
      GO TO 925
  923 CONTINUE
      IJ=IJ+1
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      FXE1=FX(IJ)
      FXE2=FX(IMJ)
      FXW1=1.-FXE1
      FXW2=1.-FXE2
      DXB=X(IJ)-X(IMJ)
      DYB=Y(IJ)-Y(IMJ)
      DXBP=QTR*(X(IJ-2)-X(IJ)+X(IMJ-2)-X(IMJ))
      DYBP=QTR*(Y(IJ-2)-Y(IJ)+Y(IMJ-2)-Y(IMJ))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
```

```
      DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
      PHI(IJ)=ED(IJ-1)-DEL*FAC
      IJ=IJ-1
      GO TO 925
  924 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNN(I))*GREAT
      BP(IJ)=GREAT
  925 CONTINUE
      AN(IJ)=0.0
  920 CONTINUE
C-----WEST BOUNDARY
      DO 930 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (931,932,933,934) JTBW(J)
  931 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 935
  932 CONTINUE
      PHI(IJ-NJ)=ED(IJ)
      GO TO 935
  933 CONTINUE
      IJ=J
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DYB=Y(IJ)-Y(IJM)
      DXB=X(IJ)-X(IJM)
      DXBP=QTR*(X(IJ+NJ)-X(IJ)+X(IJM+NJ)-X(IJM))
      DYBP=QTR*(Y(IJ+NJ)-Y(IJ)+Y(IJM+NJ)-Y(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
      PHI(IJ)=ED(IJ+NJ)-DEL*FAC
      IJ=IJ+NJ
      GO TO 935
  934 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNW(J))*GREAT
      BP(IJ)=GREAT
  935 CONTINUE
      AW(IJ)=0.0
  930 CONTINUE
C-----EAST BOUNDARY
      DO 940 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (941,942,943,944) JTBE(J)
  941 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 945
  942 CONTINUE
      PHI(IJ+NJ)=ED(IJ)
      GO TO 945
  943 CONTINUE
      IJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      FYN1=FY(IJ)
      FYN2=FY(IJM)
      FYS1=1.-FYN1
      FYS2=1.-FYN2
      DXB=X(IJ)-X(IJM)
      DYB=Y(IJ)-Y(IJM)
      DXBP=QTR*(X(IJ-NJ-NJ)-X(IJ)+X(IJM-NJ-NJ)-X(IJM))
      DYBP=QTR*(Y(IJ-NJ-NJ)-Y(IJ)+Y(IJM-NJ-NJ)-Y(IJM))
      FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
      DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
```



```
      PHI(IJ)=ED(IJ-NJ)-DEL*FAC
      IJ=IJ-NJ
      GO TO 945
  944 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNE(J))*GREAT
      BP(IJ)=GREAT
  945 CONTINUE
      AE(IJ)=0.0
  940 CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------
      SUBROUTINE AMODVIS (TE,ED,VIS,DEN,VISCOS,
     1                    ITBS,TITBN,JTBE,JTBW)
C-----------------------------------------------------------
C
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
C
      DIMENSION TE(NXNY), ED(NXNY), VIS(NXNY), DEN(NXNY)
      DIMENSION ITBS(NX), ITBN(NX), JTBE(NY), JTBW(NY)
C
      DO 10 I=1,NIM+1
      DO 10 J=1,NJM+1
         IJ=IMNJ(I)+J
         VISOLD=VIS(IJ)
         VIS(IJ)=VISCOS
         IF (ED(IJ).GT.SMALL) THEN
            VIS(IJ)=FMU(IJ)*DEN(IJ)*TE(IJ)**2*CMU/ED(IJ)
     #              +VISCOS
         ENDIF
         VIS(IJ)=URFVIS*VIS(IJ)+(1.-URFVIS)*VISOLD
   10 CONTINUE
C
      IF (LAY2) THEN
         DO 20 I=2,NIM
          DO 210 J=2,NJM
             IJ=IMNJ(I)+J
             IF(FMU(IJ).LT.0.95) VIS(IJ)=VIS2(IJ)
  210     CONTINUE
 20    CONTINUE
      END IF
C
      RETURN
      END
C
C**********************************************************
      SUBROUTINE ASOLSIP(PHI,IPHI,RESOR)
C**********************************************************
C
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
C
      DIMENSION PHI(NXNY), RES(NXNY)
      DIMENSION BS(NXNY), BN(NXNY), BE(NXNY), BW(NXNY)
c
      DIMENSION FP(NXNY), RESOR(2)
c
C
      NJ = NJM + 1
      NI = NIM + 1
      NINJ = NI*NJ
      DO 5 IJ=1,NINJ
         BN(IJ) = 0.
         BE(IJ) = 0.
         RES(IJ) = 1.
c
         FP(IJ)=PHI(IJ)
```

```
c
    5 CONTINUE
C
C

      DO 10 I=2,NIM
      DO 10 J=2,NJM
         IJ=IMNJ(I)+J
         API=1.0/AP(IJ)
         AP(IJ)=1.0
         AE(IJ)=AE(IJ)*API
         AW(IJ)=AW(IJ)*API
         AN(IJ)=AN(IJ)*API
         AS(IJ)=AS(IJ)*API
         SU(IJ)=SU(IJ)*API
   10 CONTINUE
C
      DO 20 I=2,NIM
      DO 20 J=2,NJM
         IJ=IMNJ(I)+J
         IJM=IJ-1
         IMJ=IJ-NJ
         BW(IJ)=-AW(IJ)/(1.+ALFAKE*BN(IJ-NJ))
         BS(IJ)=-AS(IJ)/(1.+ALFAKE*BE(IJM))
         POM1=ALFAKE*BW(IJ)*BN(IMJ)
         POM2=ALFAKE*BS(IJ)*BE(IJM)
         BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IMJ)-BS(IJ)*BN(IJM)
         BN(IJ)=(-AN(IJ)-POM1)/(BP(IJ)+SMALL)
         BE(IJ)=(-AE(IJ)-POM2)/(BP(IJ)+SMALL)
   20 CONTINUE
C
      DO 100 L=1,NSWPKE(IPHI)
         RESORP=0.
         DO 30 I=2,NIM
         DO 30 J=2,NJM
            IJ=IMNJ(I)+J
            RES(IJ)=AN(IJ)*PHI(IJ+1)+AS(IJ)*PHI(IJ-1)+AE(IJ)*PHI(IJ+NJ)+
     &              AW(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)
            RESORP=RESORP+ABS(RES(IJ))
            RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/
     &               (BP(IJ)+SMALL)
   30    CONTINUE
C
         IF(L.EQ.1) RESORKE(IPHI)=RESORP
         RSM=SORKE(IPHI)*RESORKE(IPHI)
         DO 40 I=2,NIM
            II=NIM+2-I
            DO 40 J=2,NJM
               JJ=NJM+2-J
               IJ=IMNJ(II)+JJ
               RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)
               PHI(IJ)=PHI(IJ)+RES(IJ)
   40    CONTINUE
co       IF(RESORP.LE.RSM) RETURN
         IF(RESORP.LE.RSM) GOTO 200
  100 CONTINUE
C
      IF(RESORP.GE.RSM.AND.L.GE.NSWPKE(IPHI)) WRITE(*,2)
    2 FORMAT(10X,' SOLSIP DID NOT CONVERGE   ')
C
  200 CONTINUE
      AUX1=0.
      AUX2=0.
      DO 50 I=2,NIM
      DO 50 J=2,NJM
      IJ=IMNJ(I)+J
      AUX1=AUX1+ABS(PHI(IJ)-FP(IJ))
      AUX2=AUX2+ABS(FP(IJ))
   50 CONTINUE
c     IF(AUX2.LT.1.E-30) AUX2=1.E-30
      RESOR(IPHI)=AUX1/AUX2
```



```
      RETURN
      END
C
C.....A GAUSS ELIMINATION SOLVER
      SUBROUTINE SOLV(A,BB,N)
      DIMENSION A(N,N),B(N),C(N),BB(N),X(N),MME(N)
      EP =1.E-19
      DO 10 J=1,N
   10 MME(J)=J
      DO 20 I=1,N
      Y=0.
      DO 30 J=I,N
      IF(ABS(A(I,J)).LT.ABS(Y)) GOTO 30
      K=J
      Y=A(I,J)
   30 CONTINUE
C
      IF(ABS(Y).LT.EP) THEN
      WRITE(*,*)
      WRITE(9,*)
      DO 35 IA=1,N
      WRITE(*,1000) (A(IA,JA),JA=1,N)
   35 CONTINUE
      PRINT*,'THERE IS NO CONVERSE MATRIX'
      STOP 2222
      ENDIF
C
      Y=1./Y
      DO 40 J=1,N
      C(J)=A(J,K)
      A(J,K)=A(J,I)
      A(J,I)=-C(J)*Y
      B(J)=A(I,J)*Y
   40 A(I,J)=A(I,J)*Y
      A(I,I)=Y
      J=MME(I)
      MME(I)=MME(K)
      MME(K)=J
      DO 11 K=1,N
      IF(K.EQ.I) GOTO 11
      DO 12 J=1,N
      IF(J.EQ.I) GOTO 12
      A(K,J)=A(K,J)-B(J)*C(K)
   12 CONTINUE
   11 CONTINUE
   20 CONTINUE
      DO 33 I=1,N
      DO 44 K=1,N
      IF(MME(K).EQ.I) GOTO 55
   44 CONTINUE
   55 IF(K.EQ.I) GOTO 33
      DO 66 J=1,N
      W=A(I,J)
      A(I,J)=A(K,J)
   66 A(K,J)=W
      IW=MME(I)
      MME(I)=MME(K)
      MME(K)=IW
   33 CONTINUE
 1000 FORMAT(4X,1P5E13.4)
      DO 50 I=1,N
      X(I)=0.
      DO 50 J=1,N
   50 X(I)=X(I)+A(I,J)*BB(J)
      DO 60 I=1,N
   60 BB(I)=X(I)
      RETURN
      END
C-----------------------------------------
      SUBROUTINE AMODIFY(SUASM,SVASM,SWASM,
     &   X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,DEN,
```

```
     &    VISCOS,TE,DNS,DNN,DNW,DNE,ITBS,ITBN,JTBW,JTBE,
     &    U,V,W,ITER)
C----------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'asm.h'
C
      DIMENSION UUW(NY),VVW(NY),UVW(NY),UWW(NY),VWW(NY)
c
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),R(NXNY)
      DIMENSION DNS(NX),DNN(NX),DNW(NY),DNE(NY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY),TE(NXNY)
      DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
c
      IF(ITER.EQ.1) THEN
      REWIND 41
      READ(41,*)
      READ(41,*)
      READ(41,*) CD1,CD2,CMU,ELOG,CAPPA
      READ(41,*)
      READ(41,*) LRE,LAY2
      END IF
C
      NI = NIM + 1
      NJ = NJM + 1
      CMU25=SQRT(SQRT(CMU))
C
C-----SOUTH BOUNDARY
      DO 600 I=2,NIM
         IJ=IMNJ(I)+2
         IF(ITBS(I).EQ.4) THEN
         DXB=X(IJ-1)-X(IJ-NJ-1)
         DYB=Y(IJ-1)-Y(IJ-NJ-1)
         ARW=SQRT(DXB**2+DYB**2)
         DXB=DXB/ARW
         DYB=DYB/ARW
         CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
         YPLS=DNS(I)*CONST/VISCOS
         IF (YPLS.LE.11.63.OR.LAY2) THEN
         TCOEF=VISCOS/DNS(I)
         ELSE
         UPLUS=LOG(ELOG*YPLS)/CAPPA
         TCOEF=CONST/UPLUS
         ENDIF
         VPINT=U(IJ)*DXB+V(IJ)*DYB
         VPINT=VPINT+W(IJ)
         VPINT=ABS(VPINT-SQRT(U(IJ-1)*U(IJ-1)+
     1           V(IJ-1)*V(IJ-1)+W(IJ-1)*W(IJ-1)))
         GENTS(I)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNS(I))
C
         ENDIF
C-----NORTH BOUNDARY
         IJ=IMNJ(I)+NJM
         IF (ITBN(I).EQ.4) THEN
         DXB=X(IJ)-X(IJ-NJ)
         DYB=Y(IJ)-Y(IJ-NJ)
         ARW=SQRT(DXB**2+DYB**2)
         DXB=DXB/ARW
         DYB=DYB/ARW
         CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
         YPLS=DNN(I)*CONST/VISCOS
         IF (YPLS.LE.11.63.OR.LAY2) THEN
         TCOEF=VISCOS/DNN(I)
         ELSE
         UPLUS=LOG(ELOG*YPLS)/CAPPA
         TCOEF=CONST/UPLUS
         ENDIF
         VPINT=U(IJ)*DXB+V(IJ)*DYB
         VPINT=VPINT+W(IJ)
         VPINT=ABS(VPINT-SQRT(U(IJ+1)*U(IJ+1)+
     1           V(IJ+1)*V(IJ+1)+W(IJ+1)*W(IJ+1)))
         GENTN(I)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNN(I))
```



```
C
         ENDIF
  600 CONTINUE
C-----WEST BOUNDARY
      DO 620 J=2,NJM
      IJ=IMNJ(2)+J
      IF (JTBW(J).EQ.4) THEN
      DXB=X(IJ-NJ)-X(IJ-NJ-1)
      DYB=Y(IJ-NJ)-Y(IJ-NJ-1)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
      YPLS=DNW(J)*CONST/VISCOS
      IF (YPLS.LE.11.63.OR.LAY2) THEN
      TCOEF=VISCOS/DNW(J)
      ELSE
      UPLUS=LOG(ELOG*YPLS)/CAPPA
      TCOEF=CONST/UPLUS
      ENDIF
      VPINT=U(IJ)*DXB+V(IJ)*DYB
      VPINT=VPINT+W(IJ)
      VPINT=ABS(VPINT-SQRT(U(IJ-NJ)*U(IJ-NJ)+
     1           V(IJ-NJ)*V(IJ-NJ)+W(IJ-NJ)*W(IJ-NJ)))
      GENTW(J)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNW(J))
      ENDIF
C-----EAST BOUNDARY
      IJ=IMNJ(NIM)+J
      IF (JTBE(J).EQ.4) THEN
      UP=U(IJ)
      VP=V(IJ)
      WP=W(IJ)
      UWALL=U(IJ+NJ)
      VWALL=V(IJ+NJ)
      WWALL=W(IJ+NJ)
      TEPR=SQRT(TE(IJ))
      DELN=DNE(J)
      RB=HAF*(R(IJ)+R(IJ-1))
      DENS=DEN(IJ)
      DXB=X(IJ)-X(IJ-1)
      DYB=Y(IJ)-Y(IJ-1)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
      YPLS=DNE(J)*CONST/VISCOS
      IF (YPLS.LE.11.63.OR.LAY2) THEN
      TCOEF=VISCOS/DNE(J)
      ELSE
      UPLUS=LOG(ELOG*YPLS)/CAPPA
      TCOEF=CONST/UPLUS
      ENDIF
      VPINT=U(IJ)*DXB+V(IJ)*DYB
      VPINT=VPINT+W(IJ)
      VPINT=ABS(VPINT-SQRT(U(IJ+NJ)*U(IJ+NJ)+
     1           V(IJ+NJ)*V(IJ+NJ)+W(IJ+NJ)*W(IJ+NJ)))
      GENTEE(J)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNE(J))
      ENDIF
  620 CONTINUE
C
C--- CALCULATE REYNOLD STRESS GRADIENT TERMS
C--- TO ADD IN THE MOMENTUM EQUATIONS.
C
      DO 302 J=2,NJM
      I=1
      IJ=IMNJ(I)+J
      UVW(J)=UV(IJ)*R(IJ)*DEN(IJ)
      UUW(J)=U2(IJ)*R(IJ)*DEN(IJ)
      VVW(J)=V2(IJ)*R(IJ)*DEN(IJ)
      UWW(J)=UW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
      VWW(J)=VW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
```

```
302   CONTINUE
C
      DO 400 I=2,NIM
      IJ=IMNJ(I)+1
      UUN=U2(IJ)*R(IJ)*DEN(IJ)
      UVN=UV(IJ)*R(IJ)*DEN(IJ)
      VVN=V2(IJ)*R(IJ)*DEN(IJ)
      UWN=UW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
      VWN=VW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
      DO 401 J=2,NJM
      IJ=IMNJ(I)+J
      UUS=UUN
      VVS=VVN
      UVS=UVN
      UWS=UWN
      VWS=VWN
      U2IJE=U2(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)
      U2IJ=U2(IJ)*R(IJ)*DEN(IJ)
      UVIJE=UV(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)
      UVIJ=UV(IJ)*R(IJ)*DEN(IJ)
      V2IJE=V2(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)
      UWIJE=UW(IJ+NJ)*R(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)
      UWIJ=UW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
      VWIJE=VW(IJ+NJ)*R(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)
      VWIJ=VW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
      U2IJN=U2(IJ+1)*R(IJ+1)*DEN(IJ+1)
      V2IJN=V2(IJ+1)*R(IJ+1)*DEN(IJ+1)
      UVIJN=UV(IJ+1)*R(IJ+1)*DEN(IJ+1)
      UWN=UW(IJ+1)*R(IJ+1)*R(IJ+1)*DEN(IJ+1)
      VWN=VW(IJ+1)*R(IJ+1)*R(IJ+1)*DEN(IJ+1)
C
      UUE=U2IJE*FX(IJ)+U2IJ*(1.-FX(IJ))
      UVE=UVIJE*FX(IJ)+UVIJ*(1.-FX(IJ))
      VVE=V2IJE*FX(IJ)+V2IJ*(1.-FX(IJ))
      UWE=UWIJE*FX(IJ)+UWIJ*(1.-FX(IJ))
      VWE=VWIJE*FX(IJ)+VWIJ*(1.-FX(IJ))
      UUN=U2IJN*FY(IJ)+U2IJ*(1.-FY(IJ))
      UVN=UVIJN*FY(IJ)+UVIJ*(1.-FY(IJ))
      VVN=V2IJN*FY(IJ)+V2IJ*(1.-FY(IJ))
      UWN=UWIJN*FY(IJ)+UWIJ*(1.-FY(IJ))
      VWN=VWIJN*FY(IJ)+VWIJ*(1.-FY(IJ))
c
      DU2NS(IJ)=UUN-UUS
      DU2EW(IJ)=UUE-UUW(J)
      DUVNS(IJ)=UVN-UVS
      DUVEW(IJ)=UVE-UVW(J)
      DV2NS(IJ)=VVN-VVS
      DV2EW(IJ)=VVE-VVW(J)
      DUWEW(IJ)=UWE-UWW(J)
      DUWNS(IJ)=UWN-UWS
      DVWEW(IJ)=VWE-VWW(J)
      DVWNS(IJ)=VWN-VWS
c
      UUW(J)=UUE
      VVW(J)=VVE
      UVW(J)=UVE
      UWW(J)=UWE
      VWW(J)=VWE
c
      RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))
      DPU2EW(IJ)=DU2EW(IJ)/RP
      DPU2NS(IJ)=DU2NS(IJ)/RP
      DPV2EW(IJ)=DV2EW(IJ)/RP
      DPV2NS(IJ)=DV2NS(IJ)/RP
c
401   CONTINUE
400   CONTINUE
c
c
      DO 500 I=2,NIM
      DO 500 J=2,NJM
```



```
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      IJP=IJ+1
      IJM=IJ-1
      DYEP=HAF*(Y(IJ)-Y(IJM)+Y(IMJ)-Y(IMJ-1))
      DYKP=HAF*(Y(IJ)-Y(IMJ)+Y(IJM)-Y(IMJ-1))
      DXKP=HAF*(X(IJ)-X(IMJ)+X(IJM)-X(IMJ-1))
      DXEP=HAF*(X(IJ)-X(IJM)+X(IMJ)-X(IMJ-1))
      DYEE=HAF*(Y(IPJ)+Y(IJ)-Y(IPJ-1)-Y(IJM))
      DXEE=HAF*(X(IPJ)+X(IJ)-X(IPJ-1)-X(IJM))
      DXKN=HAF*(X(IJP)+X(IJ)-X(IJP-NJ)-X(IMJ))
      DYKN=HAF*(Y(IJP)+Y(IJ)-Y(IJP-NJ)-Y(IMJ))
      RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))
      RPE=QTR*(R(IJ)+R(IJM)+R(IPJ)+R(IPJ-1))
      RPN=QTR*(R(IJ)+R(IMJ)+R(IJP)+R(IMJ+1))
c
c-- STRESS GRADIENT SOURCE TERM IN THE U-MOMENTUM EQUATION
      SUASM(IJ)=-(DYEP*DU2EW(IJ)-DYKP*DU2NS(IJ))-
     &            (DXKP*DUVNS(IJ)-DXEP*DUVEW(IJ))
c
c-- STRESS GRADIENT SOURCE TERM IN THE V-MOMENTUM EQUATION
      SVASM(IJ)=-(DYEP*DUVEW(IJ)-DYKP*DUVNS(IJ))-
     &            (DXKP*DV2NS(IJ)-DXEP*DV2EW(IJ))
      IF(AKSI) SVASM(IJ)=SVASM(IJ)+DEN(IJ)*W2(IJ)*VOL(IJ)/RP
c
c-- STRESS GRADIENT SOURCE TERM IN THE W-MOMENTUM EQUATION
      IF(ICAL(IRW)) THEN
      SWASM(IJ)=-(DYEP*DUWEW(IJ)-DYKP*DUWNS(IJ))-
     &            (DXKP*DVWNS(IJ)-DXEP*DVWEW(IJ))
      ENDIF
c
c
500   CONTINUE
c
c
      RETURN
      END
C
c---------------------------------------------------------------
c
c gridparam.h
C
      PARAMETER (NX=100)
      PARAMETER (NY=50)
      PARAMETER (NXNY=NX*NY)
c
c---------------------------------------------------------------
c
C asm.h
C
      PARAMETER (HAF=0.5, QTR=0.25)
      PARAMETER (SMALL=1.0E-30, GREAT=1.0E+30)
C
      COMMON /A1/ AS(NXNY), AN(NXNY), AE(NXNY), AW(NXNY),
     1            AP(NXNY), BP(NXNY), SU(NXNY),APU(NXNY),APV(NXNY)
C
      COMMON /A2/ SORKE(2), NSWPKE(2), URFKE(2),PRTKE(2)
C
      COMMON /A3/ GKE,ALFAKE,RESORKE(2),URFVIS
      COMMON /CONSTT/ CMU,CMU75,ELOG,CAPPA,
     &                C1,C2,C1W,C2W,CK,CE,CD1,CD2
      COMMON /A3/ GENTS(NX),GENTN(NX),GENTW(NY),GENTEE(NY),P11(NXNY),
     1            P22(NXNY),P33(NXNY),P12(NXNY),P13(NXNY),P23(NXNY)
      COMMON /A4/ FUNX(NXNY),FUNY(NXNY),FUNXY(NXNY),
     1            GEN(NXNY),DUDX(NXNY),DUDY(NXNY),DVDX(NXNY),
     2            DVDY(NXNY),DWDX(NXNY),DWDY(NXNY)
      COMMON /STRESS/ U2(NXNY),V2(NXNY),W2(NXNY),
     1                UV(NXNY),VW(NXNY),UW(NXNY)
C
      COMMON /WALLKE/ FLR1(NXNY), FLR2(NXNY), FMU(NXNY), VIS2(NXNY),
```

```
     1                    LRE, LAY2
      LOGICAL LAY2,LRE,AKSI,RESTART
c
c-----------------------------------------------------------------------
```

CHAPTER 5

2D/Axisymmetric Full Reynolds Stress (RSM) Turbulence Model

Table of Contents

5.1 Introduction

This report describes a self contained FORTRAN source code to compute turbulent quantities using Launder, Reece and Rodi's [1] second order closure, Reynolds stress model The module deck is designed to interface with a number of flow solvers to analyse incompressible turbulent internal flows. Detailed description of the model used is given with a special emphasis on the coupling of the mean velocity and Reynolds stresses in the discretization procedure of the generalized coordinate system using a co-located finite volume method. The module was interfaced with the REACT flow solver and tested with benchmark flows including the backward-facing step. The module was also successfully interfaced with the MAST code at the University of Alabama at Huntsville (UAH) and independently tested. The Reynolds stress model implemented produced consistently more accurate simulations that the standard k-ε model.

5.2 Theory and Model Equations

The flow is considered planar or axially symmetric, steady with constant fluid properties. Its mean field may be described by a two-dimensional time averaged equations of continuity and momentum, which can be written as;

$$\frac{\partial \rho U}{\partial x} + \frac{1}{r}\frac{\partial \rho r V}{\partial r} = 0 \tag{1}$$

$$\frac{\partial(\rho r U \Phi)}{\partial x} + \frac{\partial(\rho r V \Phi)}{\partial r} = \frac{\partial}{\partial x}\left(r\mu\frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial r}\left(r\mu\frac{\partial \Phi}{\partial r}\right) + rS_\Phi \tag{2}$$

Φ stands for any of the dependent variables, namely, U and V (axial and radial velocities respectively) and rW (radial distance r multiplied by the tangential velocity W). ρ is the fluid density, μ is the laminar viscosity. S_Φ is the source term for the variable Φ and is given by;

- Axial direction, $\Phi = U$ and $S_U = -\frac{\partial P}{\partial x} - \frac{\partial \rho \overline{u^2}}{\partial x} - \frac{1}{r}\frac{\partial \rho r\overline{uv}}{\partial r}$

- Radial direction, $\Phi = V$ and $S_V = -\frac{\partial P}{\partial r} + \frac{\rho W^2}{r} - \frac{2\mu V}{r^2} - \frac{1}{r}\frac{\partial r\overline{v^2}}{\partial r} - \frac{\partial \overline{ruv}}{\partial x} + \frac{\overline{rw^2}}{r}$

- Tangential direction, $\Phi = rW$ and $S_W = -2\frac{\mu}{r}\frac{\partial rW}{\partial r} - \rho\frac{\overline{ruw}}{\partial x} - \rho\frac{\partial r\overline{vw}}{\partial r} - 2r\overline{vw}.$

where u, v and w are the fluctuating velocity components in the axial, radial and azimuthal directions respectively.

Turbulence wall effects in the module are represented by Gibson and Launder [2] version of the high Reynolds number stress transport closure of Launder, Reece and Rodi [1]. The stress closure consists essentially of modeled transport equations for the stresses $\overline{u_i u_j}$ and for axisymmetric swirling flow it includes all the six stresses $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$, $\overline{uw}$ and $\overline{vw}$.

The set of differential equations governing the transport of Reynolds stresses ($-\overline{u_i u_j}$) is obtained from Navier-Stokes equations by multiplying the equations for the fluctuating components (u_i) and (u_j) by (u_j) and (u_i) respectively, then summing these equations and time averaging the results. The resulting Reynolds stress transport equations are then solved using the mean flow equations to obtain the mean and turbulent flow quantities.

The full transport equations for the Reynolds stresses can be written in a compact form using Cartesian tensor representation as;

$$\frac{1}{r}\frac{\partial \rho r U_k \overline{u_i u_j}}{\partial x_k} - \frac{1}{r}\frac{\partial}{\partial x_k}\left(r C_k \rho\, \overline{u_k u_l}\, \frac{k}{\varepsilon}\, \frac{\partial \overline{u_i u_j}}{\partial x_l}\right) = P_{ij} + D_{ij} + \Phi_{ij} - \varepsilon_{ij} \tag{3}$$

Where U_k are the mean velocity components in x_k-direction. The right hand side contains the production term P_{ij} given as

$$P_{ij} = -\rho\left(\overline{u_i u_k}\frac{\partial U_j}{\partial x_k} + \overline{u_j u_k}\frac{\partial U_i}{\partial x_k}\right) \tag{4}$$

P_{ij} does not require approximations since it is fully represented by turbulent stresses and mean flow gradients.

The dissipation correlation ε_{ij} arise from the fine-scale of the turbulent motion. At high Reynolds numbers these scales are many orders of magnitude smaller than the large energy containing eddies and turbulence energy cascades down along the eddy-size range with little linkage occurring at intermediate scales, to be ultimately dissipated by the smallest eddies which are unaware of the nature of the mean flow and the large scale turbulence. Therefore, the structure of these fine scale motions responsible for viscous dissipation is isotropic and the dissipation tensor ε_{ij} reduces to

$$2\,\nu\,\frac{\partial \overline{u_i u_j}}{\partial x_k \partial x_k} = \frac{2}{3}\,\varepsilon\,\delta_{ij} \tag{5}$$

An additional equations for the dissipation ε is required.

D_{ij} represents the Reynolds stress diffusion which does not in general contribute greatly to the balance of transport of $\overline{u_i u_j}$ except in regions of low stress production by mean strain. This term include contributions of fluctuating pressure-velocity correlations ($\overline{pu_i}$ and $\overline{pu_j}$), triple correlations $\overline{u_i u_j u_k}$ and viscous diffusion $\nu\,\frac{\partial \overline{u_i u_j}}{\partial x_k}$. Daly and Harlow [3] proposed a simple gradient diffusion hypothesis to model the stress diffusion term in the form

$$D_{ij} = C_s \frac{\partial}{\partial x_k}\,\rho\,\frac{k}{\varepsilon}\,[\,\overline{u_k u_l}\,\frac{\partial \overline{u_i u_j}}{\partial x_l}\,] \tag{6}$$

with constant C_s is taken to be 0.22. Lien and Leschziner [4] simplified the treatment of the diffusion term to allow an appropriate isotropic diffusivity in the form

$$D_{ij} = \frac{\partial}{\partial x_k}\,[\,\frac{\mu}{\sigma_k}\,\frac{\partial}{\partial x_k}\,(\overline{u_i u_j})] \tag{7}$$

where σ_k is a dimensionless constant. Harlow's proposal for the diffusion term is adopted in the present module since it is based on the fundamental conservation equations for the triple correlations, while Lien & Leschziner's form has a weaker basis in this respect.

Φ_{ij} represents the redistribution of turbulence energy among the normal stresses through the interaction of pressure and strain fluctuations. Modeling the pressure-strain term is the most elaborate and involves the solution of the Poisson equation for pressure fluctuations p. The explicit appearance of the pressure in the correlation is eliminated by taking the divergence of the equation for the fluctuating velocity u_i, thus obtaining a Poisson equation for p. Following a volume integration of the resulting equation subject to the assumption of local mean-flow homogeneity results in three contributions to the pressure-strain correlation Φ_{ij}. One involving just fluctuating quantities $\Phi_{ij,1}$ another arising from the presence of the mean rate of strain $\Phi_{ij,2}$. and a third arising from the surface integral representing wall effects $\Phi_{ij,w}$. Since the primary role of Φ_{ij} is to guide turbulence towards isotropy, Rotta [5] proposed for $\Phi_{ij,1}$;

$$\Phi_{ij,1} = -\,2\,\rho\,C_1\,\varepsilon\,b_{ij} \tag{8}$$

where $b_{ij} = (\overline{u_i u_j} - \frac{2}{3} \delta_{ij} k) / 2k$ is the dimensionless anisotropy parameter. C_1 is a constant and k and ε are turbulent kinetic energy and energy dissipation respectively. More elaborate models have been proposed such as Lumley [6] and Fu [7] using a nonlinear expression for $\Phi_{ij,1}$. The term $\Phi_{ij,2}$ has been the subject of more extensive research. The traditional linear approach similar to Rotta's work simplifies this correlation to:

$$\Phi_{ij,2} = - C_2 (P_{ij} - \frac{2}{3} \delta_{ij} P) \tag{9}$$

where P is the production of turbulent kinetic energy. Analogous to $\Phi_{ij,1}$ the correlation, $\Phi_{ij,2}$ represents the isotropization of turbulence production tensor with C_2 as a constant. More elaborate models such as that of Speziale, Sarkar and Gatski [8] is based on dynamical systems approach and invariancy concepts. Nonlinear models for $\Phi_{ij,2}$ based on the realizability constraints have been developed, e.g, Shih and Lumley [9] and Fu, Launder and Tselepidakis [10]. The simplified correlations in equations (8) and (9) are used in the present module.

The correlation $\Phi_{ij,w}$ represents the wall damping effects that counteracts the tendency of $\Phi_{ij,1}$ and $\Phi_{ij,2}$ to isotropise the turbulent structure. Since close to a solid wall turbulence approach a state of intense anisotropy associated with a tendency towards a 2D turbulence. Following Shir [11] and Gibson and Launder [2], $\Phi_{ij,w}$ is modeled as the combination of two separate terms;

$$\Phi_{ij,1w} = C_{1w}\, \rho\, \frac{\varepsilon}{k} \left[\overline{u_k u_m}\, n_k n_m \delta_{ij} - \frac{3}{2} \overline{u_k u_i}\, n_k n_j - \frac{3}{2} \overline{u_k u_j}\, n_k n_i \right] f\left(\frac{l}{l_n}\right) \tag{10}$$

$$\Phi_{ij,2w} = C_{2w} \left[\Phi_{km,2}\, n_k n_m \delta_{ij} - \frac{3}{2} \Phi_{ij,2}\, n_k n_j - \frac{3}{2} \Phi_{jk,2}\, n_k n_i \right] f\left(\frac{l}{l_n}\right) \tag{11}$$

where l_n is the normal distance from the point in question to the wall and $l\ (= \frac{k^{3/2}}{\varepsilon})$ is the turbulent length scale. The following relationship is used for the wall damping function

$$f = \frac{C_m^{.75} k^{3/4}}{\kappa\, \varepsilon} \frac{1}{<l_n>} \tag{12}$$

where $<l_n>$ is the average distance of the point considered from the surrounding surfaces and n_i is a wall-normal unit vector in the i -direction. The constants C_{1w} and C_{2w} have values of 0.5 and 0.3 respectively.

It will be of some value to list the full Reynolds stress equations for axisymmetric swirling flows. Although, the derivations have been carried out within the constraints of Cartesian coordinates, considerations will be given next to the forms applicable to any general curved coordinate system.

In general the transport equation for the Reynolds stresses $(\overline{u_i u_j})$ can be written as;

$$C_{ij} = D_{ij} + P_{ij} + F_{ij} - \varepsilon_{ij} + R_{ij} \tag{14}$$

where C_{ij}, D_{ij}, P_{ij}, F_{ij} and ε represent convection, diffusion, production, pressure-strain and dissipation terms. The term R_{ij} results from the transformation of the equation from plane to axially symmetric conditions and swirl. In Cartesian coordinates, the above terms are summarized below for each stress component;

• $\overline{u^2}$ - equation

$$C_{11} = \frac{1}{r}\frac{\partial}{\partial x}(\rho\, r\, U\overline{u^2}) + \frac{1}{r}\frac{\partial}{\partial r}(\rho\, r\, V\overline{u^2})$$

$$D_{11} = \frac{1}{r}\frac{\partial}{\partial x}[\, \rho\, r\, C_k \overline{u^2}\frac{k}{\varepsilon}\frac{\partial \overline{u^2}}{\partial x} + \rho\, r\, C_k \overline{uv}\frac{k}{\varepsilon}\frac{\partial \overline{u^2}}{\partial r}\,]$$

$$+ \frac{1}{r}\frac{\partial}{\partial r}[\, \rho\, r\, C_k \overline{uv}\frac{k}{\varepsilon}\frac{\partial \overline{u^2}}{\partial x} + \rho\, r\, C_k \overline{v^2}\frac{k}{\varepsilon}\frac{\partial \overline{u^2}}{\partial r}\,]$$

$$P_{11} = -2r(\,\overline{u^2}\frac{\partial U}{\partial x} + \overline{uv}\frac{\partial U}{\partial r})$$

$$\Phi_{11} = -\rho\, C_1 \frac{\varepsilon}{k}\,(\overline{u^2} - \frac{2}{3}k) - C_2\,(P_{11} - \frac{2}{3}P)$$

$$+ \rho\, C_{1w}\frac{\varepsilon}{k}[\, -2\,\overline{u^2}\, f_x + \overline{v^2}\, f_y - \overline{uv}\, f_{xy}\,]$$

$$+ C_{2w}[\, 2\, C_2 (P_{11} - \frac{2}{3}P) f_x - C_2 (P_{22} - \frac{2}{3}P) f_y + C_2 P_{12} f_{xy}\,]$$

$$\varepsilon_{11} = -\frac{2}{3}\rho\,\varepsilon$$

• $\overline{v^2}$ - equation;

$$C_{22} = \frac{1}{r}\frac{\partial}{\partial x}(\rho\, r\, U\overline{v^2}) + \frac{1}{r}\frac{\partial}{\partial r}(\rho\, r\, V\overline{v^2})$$

$$D_{22} = \frac{1}{r}\frac{\partial}{\partial x}[\,\rho\, r\, C_k \frac{k}{\varepsilon}(\overline{u^2}\,\frac{\partial \overline{v^2}}{\partial x} + \overline{uv}\,\frac{\partial \overline{v^2}}{\partial r}\,)\,] + \frac{1}{r}\frac{\partial}{\partial r}[\,\rho\, r\, C_k \frac{k}{\varepsilon}(\overline{uv}\,\frac{\partial \overline{v^2}}{\partial x} + \overline{v^2}\,\frac{\partial \overline{v^2}}{\partial y}\,)\,]$$

$$P_{22} = -\,2\,\rho\,(\,\overline{uv}\,\frac{\partial V}{\partial x} + \overline{v^2}\,\frac{\partial V}{\partial r} - \overline{vw}\,\frac{W}{r}\,)$$

$$\begin{aligned}
\Phi_{22} = {} & -\,C_1\,\rho\,\frac{\varepsilon}{k}\,(\overline{v^2} - \frac{2}{3}k\,) - C_2\,(P_{22} - \frac{2}{3}P) \\
& + \rho\, C_{1w}\,\frac{\varepsilon}{k}\,(\,\overline{u^2}\, f_x - 2\,\overline{v^2}\, f_y - \overline{uv}\, f_{xy}\,) \\
& + C_{2w}\,[\,-C_2(P_{11} - \frac{2}{3}P)\,f_x + 2\,C_2\,(P_{22} - \frac{2}{3}P\,)f_y + C_2\,P_{12}\,f_{xy}
\end{aligned}$$

$$\begin{aligned}
R_{22} = {} & 2\,C_k\,\rho\,\frac{k}{\varepsilon}\,\frac{(\overline{w^2})^2}{r^2} - \frac{2}{r}\frac{\partial}{\partial r}(\,\rho\, C_k\,\frac{k}{\varepsilon}\,(\overline{vw})^2) - 2\,\frac{\partial}{\partial x}(\,\rho\, C_k\,\frac{k}{\varepsilon}\,\overline{uw}\,\frac{\overline{vw}}{r}\,) \\
& -\,2\,\rho\, C_k\,\frac{k}{\varepsilon}\,\frac{\overline{w^2}}{r^2}\,\overline{v^2} - 2\,\rho\, C_k\,\frac{k}{\varepsilon}\,\frac{\overline{uw}}{r}\,\frac{\partial \overline{vw}}{\partial x} - 2\,\rho\, C_k\,\frac{k}{\varepsilon}\,\frac{\overline{vw}}{r}\,\frac{\partial \overline{vw}}{\partial r} + 2\,\rho\,\overline{vw}\,\frac{W}{r}
\end{aligned}$$

•$\overline{w^2}$ - equation

$$C_{33} = \frac{1}{r}\frac{\partial}{\partial x}(\rho\, r\, U\overline{w^2}) + \frac{1}{r}\frac{\partial}{\partial r}(\rho\, r\, V\overline{w^2})$$

$$\begin{aligned}
D_{33} = {} & \frac{1}{r}\frac{\partial}{\partial x}[\,\rho\, r\, C_k\,\frac{k}{\varepsilon}\,(\,\overline{u^2}\frac{\partial \overline{w^2}}{\partial x} + \overline{uv}\,\frac{\partial \overline{w^2}}{\partial r})\,] \\
& + \frac{1}{r}\,\frac{\partial}{\partial r}[\,\rho\, r\, C_k\,\frac{k}{\varepsilon}\,(\,\overline{uv}\,\frac{\partial \overline{w^2}}{\partial x} + \overline{v^2}\,\frac{\partial \overline{w^2}}{\partial r})\,]
\end{aligned}$$

$$P_{33} = -\,2\,\rho\,(\overline{uw}\,\frac{\partial W}{\partial x} + \overline{vw}\,\frac{\partial W}{\partial r} + \overline{w^2}\,\frac{V}{r}\,)$$

$$\begin{aligned}
F_{33} = {} & -\,\rho\, C_1\,\frac{\varepsilon}{k}\,(\overline{w^2} - \frac{2}{3}k) - C_2\,(P_{33} - \frac{2}{3}P) \\
& + \rho\, C_{1}w\,\frac{\varepsilon}{k}\,[\,\overline{u^2}\, f_x + \overline{v^2}\, f_y + 2\,\overline{uv}\, f_{xy}\,] \\
& -\,C_2\, C_{2w}\,[\,(P_{11} - \frac{2}{3}P\,)\,f_x + (P_{22} - \frac{2}{3}P)\,f_y - 2\,C_2 P_{12} f_{xy}\,]
\end{aligned}$$

$$R_{33} = 2\rho C_k\,\frac{k}{\varepsilon}\,\overline{w^2}\,\frac{\overline{v^2}}{r^2} + \frac{2}{r}\frac{\partial}{\partial r}(\,\rho C_k\,\frac{k}{\varepsilon}\,(\overline{vw})^2\,) + 2\,\frac{\partial}{\partial x}(\,\rho C_k\,\frac{k}{\varepsilon}\,\overline{uw}\,\frac{\overline{vw}}{r}\,)$$

$$- 2\,\rho\,C_k\,\frac{k}{\varepsilon}\frac{(\overline{w^2})^2}{r^2} + 2\,\rho\,C_k\,\frac{k}{\varepsilon}\frac{\overline{uw}}{r}\frac{\partial\overline{vw}}{\partial x} + 2\,\rho\,C_k\,\frac{k}{\varepsilon}\frac{1}{r}\,\overline{vw}\,\frac{\partial\overline{vw}}{\partial r} - 2\rho\,\overline{vw}\,\frac{W}{r}$$

• $\overline{uv}$ - equation

$$C_{12} = \frac{1}{r}\frac{\partial}{\partial x}(\,\rho r U\,\overline{uv}) + \frac{1}{r}\frac{\partial}{\partial r}(\rho r V\,\overline{uv})$$

$$D_{12} = \frac{1}{r}\frac{\partial}{\partial x}[\,\rho r C_k\frac{k}{\varepsilon}(\,\overline{u^2}\,\frac{\partial\overline{uv}}{\partial x} + \overline{uv}\,\frac{\partial\overline{uv}}{\partial r}\,)\,] + \frac{1}{r}\frac{\partial}{\partial r}[\,\rho r C_k\frac{k}{\varepsilon}(\,\overline{uv}\,\frac{\overline{uv}}{\partial x} + \overline{v^2}\,\frac{\partial\overline{uv}}{\partial r}\,)\,]$$

$$P_{12} = -\,\rho\,(\,\overline{u^2}\,\frac{\partial V}{\partial x} - \overline{uv}\,\frac{V}{r} + \overline{v^2}\,\frac{\partial U}{\partial r} - \overline{uw}\,\frac{W}{r}\,)$$

$$\Phi_{12} = -\,\rho\,C_1\,\frac{\varepsilon}{k}\,\overline{uv} - C_2\,P_{12}$$
$$-\,\frac{3}{2}\rho C_{1w}\,\frac{\varepsilon}{k}\,[\,\overline{uv}(f_x + f_y) + (\overline{u^2} + \overline{v^2})\,f_{xy}\,]$$
$$+\,\frac{3}{2}\,C_2\,C_{2w}\,[\,(P_{11} + P_{22} - \frac{4}{3}P)\,f_{xy} + P_{12}\,(f_x + f_y)\,]$$

$$R_{12} = -\,\rho\,C_k\,\frac{k}{\varepsilon}\,\overline{w^2}\,\frac{\overline{uv}}{r^2} - \frac{\partial}{\partial x}(\,\rho\,C_k\,\frac{k}{\varepsilon}\frac{(\overline{uw})^2}{r}\,) - \frac{1}{r}\frac{\partial}{\partial r}(\,\rho\,C_k\,\frac{k}{\varepsilon}\,\overline{uw}\;\overline{vw}\,)$$
$$-\,\rho\,C_k\,\frac{1}{r}\frac{k}{\varepsilon}(\,\overline{uw}\,\frac{\partial\overline{uw}}{\partial x} + \overline{vw}\,\frac{\partial\overline{uw}}{\partial r}\,) + \rho\,\overline{uw}\,\frac{W}{r}$$

• $\overline{vw}$ - equation

$$C_{23} = \frac{1}{r}\frac{\partial}{\partial x}(\,\rho\,r\,U\,\overline{vw}) + \frac{1}{r}\frac{\partial}{\partial r}(\,\rho\,r\,V\,\overline{vw}\,)$$

$$D_{23} = \frac{1}{r}\frac{\partial}{\partial x}[\,\rho\,r\,C_k\frac{k}{\varepsilon}(\,\overline{u^2}\,\frac{\partial\,\overline{vw}}{\partial x} + \overline{uv}\,\frac{\partial\,\overline{vw}}{\partial r})\,]$$
$$+\,\frac{1}{r}\frac{\partial}{\partial r}[\,\rho\,r\;C_k\frac{k}{\varepsilon}(\,\overline{uv}\,\frac{\partial\,\overline{vw}}{\partial x} + \overline{v^2}\,\frac{\partial\,\overline{vw}}{\partial r}\,)\,]$$

$$P_{23} = -\,\rho\,[\,\overline{uv}\,\frac{\partial W}{\partial x} + \overline{uw}\,\frac{\partial V}{\partial x} + \overline{v^2}\,\frac{\partial W}{\partial r} + \overline{vw}\,\frac{\partial V}{\partial r} + \overline{vw}\,\frac{V}{r} - \overline{w^2}\,\frac{W}{r}\,]$$

$$\Phi_{23} = -\rho C_1 \frac{\varepsilon}{k} \overline{vw} - C_2 P_{23} - \frac{3}{2} \rho C_{1w} \frac{\varepsilon}{k} (\overline{uw} f_{xy} + \overline{vw} f_y) + \frac{3}{2} C_2 C_{2w} (P_{13} f_{xy} + P_{23} f_y)$$

$$R_{23} = -\rho (\overline{v^2} - \overline{w^2}) \frac{W}{r} + \rho C_k \frac{k}{\varepsilon} \frac{1}{r} \overline{vw} \frac{\partial}{\partial r} (\overline{v^2} - \overline{w^2}) + \frac{\partial}{\partial x} (\rho C_k \frac{k}{\varepsilon} \overline{uw} \frac{(\overline{v^2} - \overline{w^2})}{r})$$

$$+ \frac{1}{r} \frac{\partial}{\partial r} (\rho C_k \frac{k}{\varepsilon} \overline{vw} (\overline{v^2} - \overline{w^2})) - 4 \rho C_k \frac{k}{\varepsilon} \frac{1}{r} \overline{vw} \frac{\overline{w^2}}{r} + \rho C_k \frac{1}{r} \frac{k}{\varepsilon} \overline{uw} \frac{\partial}{\partial x} (\overline{v^2} - \overline{w^2})$$

•$\overline{uw}$-equation

$$C_{13} = \frac{1}{r} \frac{\partial}{\partial x} (\rho r U \overline{uw}) + \frac{1}{r} \frac{\partial}{\partial r} (\rho r V \overline{uw})$$

$$D_{13} = \frac{1}{r} \frac{\partial}{\partial x} [\rho r C_k \frac{k}{\varepsilon} (\overline{u^2} \frac{\partial \overline{uw}}{\partial x} + \overline{uv} \frac{\partial \overline{uw}}{\partial r})]$$

$$+ \frac{1}{r} \frac{\partial}{\partial r} [\rho r C_k \frac{k}{\varepsilon} (\overline{uv} \frac{\partial \overline{uw}}{\partial x} + \overline{v^2} \frac{\partial \overline{uw}}{\partial r})]$$

$$P_{13} = -\rho (\overline{u^2} \frac{\partial W}{\partial x} + \overline{uv} \frac{\partial W}{\partial r} + \overline{vw} \frac{\partial U}{\partial r} - \overline{uw} \frac{\partial V}{\partial r})$$

$$F_{13} = -\rho C_1 \frac{\varepsilon}{k} \overline{uw} - C_2 P_{13} - \frac{3}{2} \rho C_{1w} \frac{\varepsilon}{k} \overline{uw} f_x - \frac{3}{2} \rho C_{1w} \frac{\varepsilon}{k} \overline{vw} f_{xy} + \frac{3}{2} C_{2w} C_2 P_{13} f_x + \frac{3}{2} C_2 C_{2w} P_{23} f_{xy}$$

$$R_{13} = - \rho \overline{uv} \frac{W}{r} + \rho C_k \frac{k}{\varepsilon} \frac{1}{r} \overline{vw} \frac{\partial \overline{uv}}{\partial r} + \frac{\partial}{\partial x} (\rho C_k \frac{k}{\varepsilon} \overline{uw} \frac{\overline{uv}}{r})$$

$$+ \frac{1}{r} \frac{\partial}{\partial r} (\rho C_k \frac{k}{\varepsilon} \overline{vw} \overline{uv}) + \rho C_k \frac{k}{\varepsilon} \frac{\overline{uw}}{r} \frac{\partial \overline{uv}}{\partial x} - \rho C_k \frac{k}{\varepsilon} \overline{w^2} \frac{\overline{uw}}{r^2}$$

The turbulence energy dissipation rate ε is determined from its own transport equation;

$$\frac{1}{r} \frac{\partial \rho r U_k \varepsilon}{\partial x_k} = \frac{1}{r} \frac{\partial}{\partial x_k} (r C_\varepsilon \rho \frac{k}{\varepsilon} \overline{u_k u_l} \frac{\partial \varepsilon}{\partial x_l}) + C_{\varepsilon 1} \frac{\varepsilon}{k} P_k - C_{\varepsilon 2} \rho \frac{\varepsilon^2}{k} \qquad (14)$$

where the constants $C_{\varepsilon 1}$ and $C_{\varepsilon 1}$ have values of 1.44 and 1.92 respectively.

The terms f_x, f_y and f_{xy} appearing in the stress-equation are tied to the orientation of the wall through the wall-damping function f and will be explained later in the wall reflection treatment section.

5.3 Boundary Conditions

To solve the transport equations for the Reynolds stresses, boundary conditions for the stresses are needed. In the present module the log-law based relations are used to bridge the gap between the fully turbulent and viscous near-wall regions. Boundary values for the stresses can be derived by applying the Reynolds stress equations to the near-wall equilibrium flow. It can be shown that the stresses are related to the turbulent kinetic energy $\overline{u_i u_j} = C_{ij}\, k$, where C_{ij} are constants to be determined. Consider as an example, the log-layer turbulent flow, where $S = \frac{\partial U}{\partial y} = \frac{u_\tau}{\kappa y}$, where u_τ is the friction velocity and κ is Von Karman constant. In the log-layer, the limiting form of the stress equation is obtained by neglecting the convective terms and equating the production to dissipation and setting the wall-distance function $f = 1$, hence the molecular and turbulent diffusion terms can be neglected. Consequently, the normal stress equation for the wall-normal component when simplified with $\Phi_{22} = \rho \frac{2}{3} \varepsilon$ is ;

$$\frac{\widetilde{v^2}}{k} = \frac{2\,(-1 + C_1 + C_2 - 2C_2\, C_{2w})}{3\,(C_1 + 2C_{1w})} = C_{22} \tag{15}$$

From experimental data, Lien & Leschziner [4] reported a value of $C_{22} \sim 0.249$ for near wall equilibrium turbulence. The most frequently used value of $C_1 = 1.8$ and $C_2 = 0.6$, and from Gibson and Launder [2] $C_{1w} = 0.5$ and $C_{2w} = 0.3$. Substituting these values into equation (13) give a value $C_{22} = 0.247$ which is close to the experimental value. Similarly, these constants also give $C_{11} = 1.09$, $C_{33} = 0.654$ and $C_{12} = -0.255$.

5.4 Numerical Procedure

The conservation equations for the Reynolds stresses and the energy dissipation are integrated over control volumes after transformation of the Cartesian form to body-fitted no-orthogonal coordinates. The equation governing the transport of a scalar property Φ, which stands for the Reynolds stress components and the energy dissipation equation can be written as;

$$\frac{1}{J}\frac{\partial}{\partial \xi^k}\left[J\left(\rho U_m \Phi - q_m\right) \beta_m^k \right] = S^{\Phi} \tag{16}$$

where ξ^k represents the curvilinear coordinate frame and J is the Jacobian of the coordinate transformation, and β_m^k represents its cofactors and qm represents the diffusion flux. Equation (16) is then integrated over discrete control volumes where the dependent variables on the volume faces are approximated by finite-difference representation.

In general the diffusion term is represented as

$$q_m = \Gamma_{\Phi} \frac{\partial \Phi}{\partial \xi^n} \beta_l^n \tag{17}$$

where Γ_{Φ} is the diffusion coefficient.

The tensorial form of the diffusivity due to Daly and Harlow [3] is adopted as;

$$\Gamma_{\Phi} = \rho\, r\, C_s \frac{k}{\varepsilon} \overline{u_m u_l} \tag{18}$$

instead of the isotropic diffusivity ($\Gamma_{\Phi} = \mu_t / \sigma_{\Phi}$). Utilizing the equilibrium assumption and experimental near-wall stress data, the constant C_s is taken to be 0.22 for the Reynolds stress equations and 0.18 for the turbulent energy dissipation equation. The diffusion term is discretized with a second-order central differencing scheme, while the convective terms are discretized using first or second order upwind differencing scheme.

A special discretization practice for the Reynolds stress gradients is introduced into the finite volume procedure with colocated storage arrangement. This is necessary to avoid the problem of mean velocity-Reynolds stress decoupling that can lead to oscillatory solutions or even divergence of the iterative solution algorithm. The procedure adopted in the present work differs from that of Obi & Peric [12] and that of Lien and Leschziner [4] by accounting for all the driving forces of the Reynolds stresses and not only those given by the gradient-diffusion type process. To illustrate the origin of the problem, consider the Reynolds stress gradient terms in the axial momentum equation in 2D Cartesian uniform grid for simplicity

$$-\frac{\partial \overline{u^2}}{\partial x} - \frac{\partial \overline{uv}}{\partial y}$$

Now integrating over a control volume surrounding node P (cf. Figure 1) yields;

$$-\int\int \frac{\partial \overline{u^2}}{\partial x}\, dx\, dy - \int\int \frac{\partial \overline{uv}}{\partial y}\, dx\, dy = -\left[(\overline{u^2}_e - \overline{u^2}_w)\, \Delta y_P + (\overline{uv}_n - \overline{uv}_s)\, \Delta x_P \right]$$

Now if the cell face values of the $\overline{u^2}_e$, $\overline{u^2}_w$, $\overline{uv}_n$ and $\overline{uv}_s$ are evaluated with linear interpolation, the stress difference expression become

$$-\left[\left(\frac{\overline{u^2}_E - \overline{u^2}_W}{2}\right) \Delta y_P + \left(\frac{\overline{uv}_N - \overline{uv}_S}{2}\right) \Delta x_P \right]$$

and since no P -node shear stress appear in the resulting expression, a chequer-board oscillation, similar to that played by the pressure field appear. Therefore, a non-linear interpolation scheme is needed to avoid these odd-even oscillations in the same context of Rhie and Chow [13] for cell face velocities. This means that any cell-face velocity is not merely sensitized to the pressure differences centered on that face but also to the Reynolds stress differences.

Consider the descretized equation for the axial normal stress component $\overline{u^2}$ in general non-orthogonal coordinates;

$$A_p\, \overline{u^2}_P = \sum_{i=n} A_i\, \overline{u^2}_i + S_{\overline{u^2}} \tag{19}$$

where n stands for the cells E, W, N and S neighboring P, A_i are the coefficients for the neighboring cells and $S_{\overline{u^2}}$ is the source term that includes production, dissipation and pressure-strain redistribution terms as;

$$S_{\overline{u^2}} = P_{11} - \frac{2}{3}\rho\,\varepsilon + \Phi_{11}$$

where Φ_{11} combines Rotta's stress isotropization model and isotropization of production model and related wall-correction terms due to Gibson and Launder [2].

$$\begin{aligned}\Phi_{11} = \; & -\rho\, C_1 \frac{\varepsilon}{k}\left(\overline{u^2} - \frac{2}{3}k\right) - C_2\left(P_{11} - \frac{2}{3}P\right) \\ & + \rho\, C_{1w}\frac{\varepsilon}{k}\left(-2\overline{u^2} f_x + \overline{v^2} f_y - \overline{uv} f_{xy}\right) \\ & + 2\, C_2\, C_{2w}\left(P_{11} - \frac{2}{3}P\right) f_x - C_2\, C_{2w}\left(P_{22} - \frac{2}{3}P\right) f_y + C_2\, C_{2w}\, P_{12} f_{xy}\end{aligned} \tag{20}$$

Rearranging the production terms that contribute to the stress generation and noting that

$P = \frac{1}{2} P_{kk}$, then;

$$S_{\overline{u^2}} = AP_{11} + BP_{22} + CP_{33} + DP_{12} + S_{11} \tag{21}$$

where

$$A = 1 - \frac{2}{3} C_2 + \frac{1}{3} C_2 C_{2w} f_y + \frac{4}{3} C_2 C_{2w} f_x$$

$$B = 1 - \frac{2}{3} C_2 + \frac{1}{3} C_2 C_{2w} f_y + \frac{4}{3} C_2 C_{2w} f_x$$

$$C = \frac{1}{3} C_2 + \frac{1}{3} C_2 C_{2w} (f_y - 2f_x)$$

$$D = C_2 C_{2w} f_{xy}$$

and S_{11} contains the remaining terms.

Substituting for the production terms P_{11}, P_{22}, P_{33} and P_{12}, then equation (19) becomes;

$$\begin{aligned}
\overline{u^2}_P = \; & H_P + 2\rho A \,[\, \overline{u^2} \,(D_1 \Delta U^{\xi} + D_2 \Delta U^{\eta}) + \overline{uv} \,(E_1 \Delta U^{\eta} + E_2 \Delta U^{\xi}) \,]_P \\
& + 2\rho B \,[\, \overline{uv} \,(D_1 \Delta V^{\xi} + D_2 \Delta V^{\eta}) + \overline{v^2} \,(E_1 \Delta V^{\xi} + E_2 \Delta V^{\xi}) + \frac{\overline{vw}}{A_p} \frac{W}{r} \,]_P \\
& + 2\rho C \,[\, \overline{uw} \,(D_1 \Delta W^{\xi} + D_2 \Delta W^{\eta}) + \overline{vw} \,(E_1 \Delta W^{\xi} + E_2 \Delta W^{\xi}) - \frac{\overline{w^2}}{A_P} \frac{W}{r} \,]_P \\
& + \rho D \,[\, \overline{u^2} \,(D_1 \Delta V^{\xi} + D_2 \Delta V^{\eta}) + \overline{v^2} \,(E_1 \Delta U^{\eta} + E_2 \Delta U^{\xi}) + \frac{\overline{uv}}{A_p} \frac{V}{r} + \frac{\overline{uw}}{A_P} \frac{W}{r} \,]_P \\
& + \frac{S_{11}}{A_p}
\end{aligned} \tag{22}$$

where

$$H_P = \Sigma_{i=n} A_i \overline{u^2}_i / A_P$$

$$D_1 = - \Delta y_P^{\eta} / A_P, \; D_2 = \Delta y_P^{\xi} / A_P,$$

$$E_1 = - \Delta x_P^{\xi} / A_P \quad \text{and} \quad E_2 = \Delta x_P^{\eta} / A_P$$

here $\Delta y_P^{\eta} = (y_n - y_s)$, $\Delta y_P^{\xi} = (y_e - y_w)$, etc

and $\Delta U^{\xi} = (U_E - U_P)$, $\Delta V^{\xi} = (V_E - V_P)$, etc

Now, performing the interpolation practice to obtain east cell-face value of the normal stress ($\overline{u^2}_e$) we obtain;

$$\begin{aligned}
\overline{u^2}_e = &< \overline{u^2}_P > - < 2\rho A\overline{u^2} D_1\, \Delta U^{\xi} > - < 2\rho A\overline{uv}\, E_2\, \Delta U^{\xi} > \\
&- < 2\rho B\overline{uv}\, D_1\, \Delta V^{\xi} > - < 2\rho B\overline{v^2} E_2\, \Delta V^{\xi} > \\
&- < 2\rho C\overline{uw} D_1\, \Delta W^{\xi} > - < 2\rho C\overline{vw} E_2\, \Delta W^{\xi} > \\
&- < \rho D\overline{u^2} D_1\, \Delta V^{\xi} > - < \rho D\, \overline{v^2}\, E_2\, \Delta U^{\xi} > \\
&+ < 2\rho A\overline{u^2} D_1 > \Delta U^{\xi} + < 2\rho A\, \overline{uv} E_2 > \Delta U^{\xi} \\
&+ 2\rho B\, \overline{uv}\, D_1 > \Delta V^{\xi} + < 2\rho V\, \overline{v^2} E_2 > \Delta V^{\xi} \\
&+ < 2\rho C\, \overline{uw} D_1 > \Delta W^{\xi} + < 2\rho C\overline{vw}\, E_2 > \Delta W^{\xi} \\
&+ < 2\rho D\, \overline{u^2}\, D_1 > \Delta V^{\xi} + < \rho D\, \overline{v^2}\, E_2 > \Delta U^{\xi}
\end{aligned} \tag{23}$$

The brackets < and > denote linear interpolation. For instance, on the east face

$$< \Phi > = (1-f_\xi)\, \Phi_P + f_\xi\, \Phi_E \quad \text{where, } f_\xi = \frac{\Delta x_P}{\Delta x_P + \Delta x_E}$$

Similar expressions can be obtained for $\overline{u^2}_w$,, $\overline{u^2}_n$, $\overline{u^2}_s$, $\overline{uv}_w$, $\overline{uv}_n$, and $\overline{uv}_s$ which are then used to calculate the Reynolds stress gradients in the discretized axial momentum equation. Similarly, expressions for $\overline{uv}_e$, $\overline{uv}_w$, , $\overline{uv}_n$, $\overline{uv}_s$, ,$\overline{v^2}_e$, $\overline{v^2}_w$, $\overline{v^2}_s$ and $\overline{v^2}_n$ can be obtained for the stress gradients in the radial momentum equation and $\overline{uw}_e$, $\overline{uw}_w$, $\overline{uw}_n$, $\overline{uw}_s$, $\overline{vw}_e$, $\overline{vw}_w$, $\overline{vw}_n$ and $\overline{vw}_s$ expressions to evaluate stress gradients in the azimuthal momentum equation.

5.4.1 Wall Reflection Treatment

The wall reflection terms $\Phi_{ij,w}$ appear in the pressure-strain term correlation as wall correction terms ($\Phi_{ij,1w}$ and $\Phi_{ij,2w}$) to counteract the tendency of $\Phi_{ij,1}$ and $\Phi_{ij,2}$ to isotropise the turbulence structure. Special consideration is given to the wall proximity effects on the redistribution process $\Phi_{ij,w}$ with relation to the local orthogonal coordinate system at the wall, cf. figure 2.

At a wall, turbulence approach a state of strong anisotropy associated with the tendency towards a 2D turbulence. The wall-reflection terms ensure that normal stress normal to the wall is not too high. For body-fitted coordinates, there is a need to consider the tensorial form of the wall reflection terms since they are tied to the orientation of the wall through the damping function term (eq. 12). For a curved surface (figure 2), the wall normal vector $\boldsymbol{n} = n_1\boldsymbol{i}_1 + n_2\boldsymbol{i}_2$, where $\boldsymbol{i}_1$ and $\boldsymbol{i}_2$ are unit vectors in Cartesian coordinates. The Cartesian components of the wall-distance function f are given as;

$$f_x = n_1^2 f, \quad f_y = n_2^2 f \quad \text{and} \quad f_{xy} = n_1 n_2 f$$

where $f_x = n_1^2\,(C_\mu^{0.75}\,k^{1.5}/\,\kappa\,\varepsilon)\,/\,L_n$ for example and L_n is the normal distance from the wall.

The Reynolds stresses close to the wall are transformed from wall coordinates to Cartesian coordinates by appropriate vector decompositioning to give;

$$\overline{u^2} = \widetilde{\overline{u^2}}\,t_1^2 + \widetilde{\overline{v^2}}\,n_1^2 + 2\,\widetilde{\overline{uv}}\,t_1 n_1$$

$$\overline{v^2} = \widetilde{\overline{u^2}}\,t_2^2 + \widetilde{\overline{v^2}}\,n_2^2 + 2\,\widetilde{\overline{uv}}\,t_2 n_2$$

$$\overline{w^2} = \widetilde{\overline{w^2}}$$

$$\overline{uv} = \widetilde{\overline{u^2}}\,t_1\,t_2 + \widetilde{\overline{v^2}}\,n_1 n_2 + \widetilde{\overline{uv}}\,(\,t_1\,n_2 + t_2\,n_1\,)$$

$$\overline{vw} = \widetilde{\overline{uw}}\,t_2 + \widetilde{\overline{vw}}\,n_2$$

$$\overline{uw} = \widetilde{\overline{uw}}\,t_1 + \widetilde{\overline{vw}}\,n_1$$

where $\overline{u^2}$, $\overline{v^2}$ .... are the Reynolds stresses in Cartesian coordinates and $\widetilde{\overline{u^2}}$, $\widetilde{\overline{v^2}}$..... are the Reynolds stresses in wall-coordinate, n_1, n_2 are the Cartesian components of the normal vector component and t_1, t_2 are the Cartesian components of the tangential vector component.

5.5 Module Evaluation

The RSM module was tested at Rocketdyne after interfacing with the CFD solver REACT and at the University of Alabama at Huntsville (UAH) using own solver (MAST). The first test was on fully developed channel flow with length to height ratio of 50 and a Reynolds number of $2\text{x}10^5$ based on the channel height. A non-uniform mesh of 101x41 was used with clustering at the walls. Figure 3 shows the fully developed mean velocity profile across the channel. Figure 4 shows the normal Reynolds stress profiles across the channel and figure 5 shows the shear stress profile. Similar results were obtained when the module was interfaced and tested independently at UAH using the MAST code.

The next test problem is that of the backward facing step of Driver and Seegmiller [14]. The calculations were performed using a 101x41 grid points with clustering near the walls. The computational domain had a length of 50H (H is the step height) and a width of 9H. The experimental data were used to specify the inflow conditions for a channel flow calculation where the fully developed profiles at the channel exit were used as the inlet conditions for the backward facing step calculations. Fully developed flow conditions were imposed at the outflow boundary. The boundary conditions for the Reynolds stress equations were arrived at by using the log-law of the wall and assuming local equilibrium conditions close to the wall. It can be shown that the Reynolds stresses are related to the turbulent kinetic energy by;

$$\overline{u_i u_j} = C_{ij}\, k \qquad (24)$$

were C_{ij} is constant. The Reynolds stresses at the vicinity of the wall used are

$$\overline{u^2} = 1.098\, k,\quad \overline{v^2} = 0.247\, k,\quad \overline{uv} = -\,0.255\, k \quad \text{and} \quad \overline{w^2} = 2k - \overline{u^2} - \overline{v^2} = 0.654\, k.$$

Figure 6 shows the stream lines using Launder, Reece and Rodi's model. The computed reattachment length was about 5.8H which is closer to the experimental value of 6.1H than the standard k-ε model (5.35H). The figure also shows a small (turbulence driven) secondary flow region at the base corner of the step which cannot be predicted using the isotropic eddy-viscosity k-ε model. Also, a smaller recirculation region is noted at the top lip of the step which is also driven by turbulence anisotropy (more refined grid may be needed to isolate and study this region). Figure 7, shows the mean velocity profile across the channel at four step heights downstream of the step as compared with the standard k-ε turbulence model predictions. The axial normal turbulent intensity ($\overline{u^2}$) profile across the channel at x/H=4 is shown on figure 8 and the radial normal turbulent intensity ($\overline{v^2}$) is shown on figure 9. The shear stress ($\overline{uv}$) profile across the channel at x/H=4 is also shown on figure 10. The results show that the module predicts improved results using the RSM model as compared with the standard k-ε model.

REFERENCES

[1] Launder, B. E., Reece, G. J. and Rodi, W. "Progress in the Development of a Reynolds Stress Closure" J. Fluid Mech. vol. 68, pp.537- , 1975.

[2] Gibson, M. and Launder, B. E. "Ground Effects on Pressure Fluctuations in the Atmospheric Boundary Layer" J. Fluid Mech., vol. 86, pt. 3, pp. 491- , 1978.

[3] Daly, B. and Harlow, F. "Transport Equations in Turbulence" Phys. Fluids, vol. 13, pp.2634- , 1970.

[4] Lien, F.-S. and Leschziner, M. A. "Second-Moment Modelling of Recirculating Flow With a Non-Orthogonal Finite-Volume Algorithim" Turbulent Shear Flows 8, Springer Verlag, pp. 205-222, 1993.

[5] Rotta, J. C "Statistische Theorie Nichthomogener Turbulenz" Z. Phys. vol. 129, pp.547- , 1951.

[6] Lumley, J. L. "Computational Modelling of Turbulent Flows" Adv. Appl. Mech., vol. 18, pp.123- , 1978.

[7] Fu, S. "Computational Modelling of Turbulent Swirling Flows With Second-Moment Closure" Ph.D. Thesis, University of Manchester, 1988.

[8] Speziale, C. G, Sarkar, S. and Gatski, T. B. "Modelling the Pressure-Strain Correlation of Turbulence: An Invariant Dynamical Systems Approach" J. Fluid Mech., vol. 227, pp. 245-272, 1991.

[9] Shih, T. -H. and Lumley, J. L. "Modeling of Pressure Correlation Terms in Reynolds Stress and Scalar Flux Equations" Tec. Rep. FDA-85-3, Cornell University, 1985.

[10] Fu, S., Launder, B. E. and Tselepidakis, D. P. "Accounting the Effects of High Strain Rates in Modeling the Pressure-Strain Corellation" UMIST Mech.Engng. Dept. Rep. TFD/97/5, 1987.

[11] Shir, C. C. "A Preliminary Numerical Study of Atmospheric Turbulent Flows in the Idealized Planetary Boundary Layer" J. Atmos. Sci., vol. 30, pp. 1327-1339, 1973.

[12] Obi, D. and Peric, M. "Second-Moment Calculations Procedure for Turbulent Flows With Collocated Variable Arrangement" AIAA J., vol. 29, no. 4, pp. 585- , 1991.

[13] Rhie, C. M. and Chow, W. L. "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation" AIAA J., vol. 21, pp. 1525- , 1983.

[14] Driver, D. and Seegmiller, H. "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow" AIAA J., vol. 23, pp. 163-171, 1985.

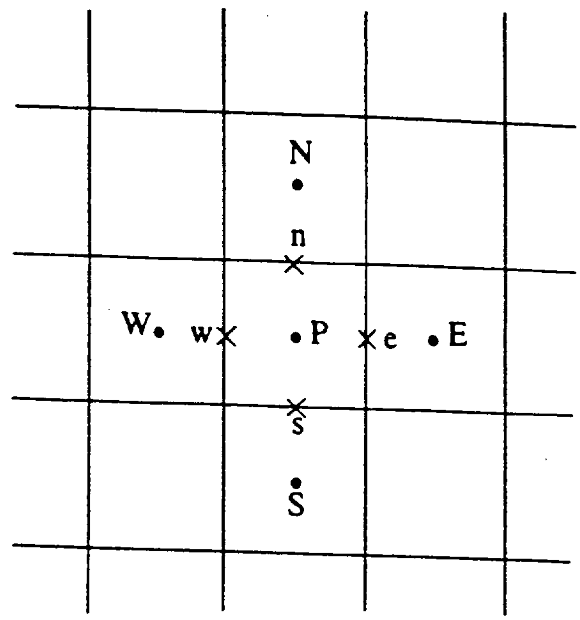


Figure 1. Control volume for node P and it's surroundings

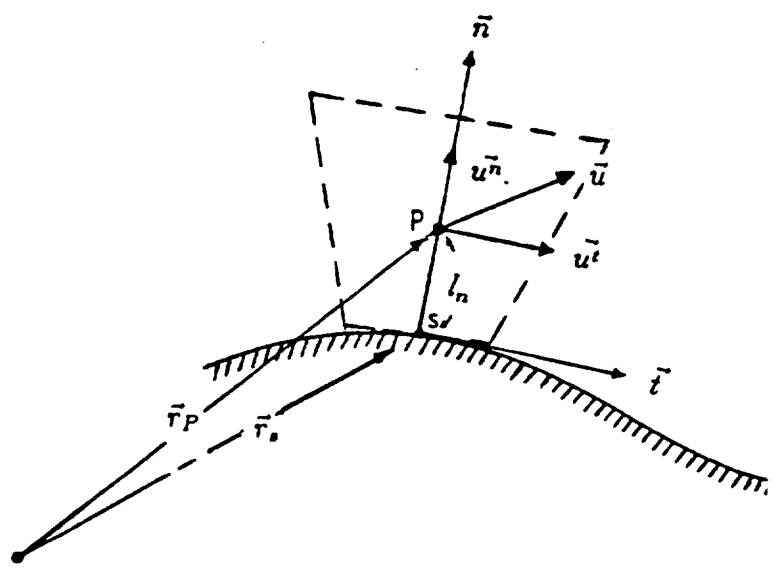


Figure 2. Cartesian and wall-coordinate systems

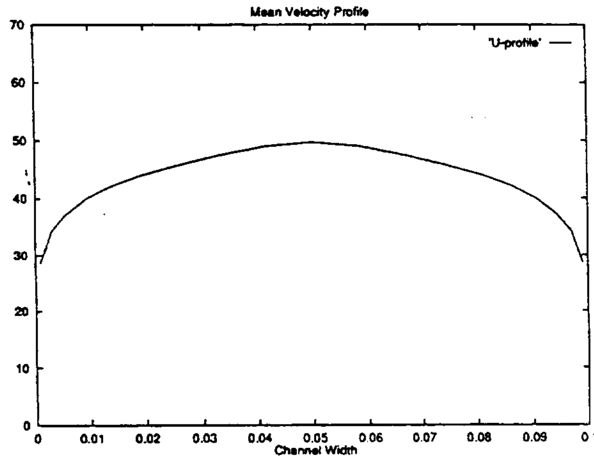


Figure 3. Mean velocity profile across the channel

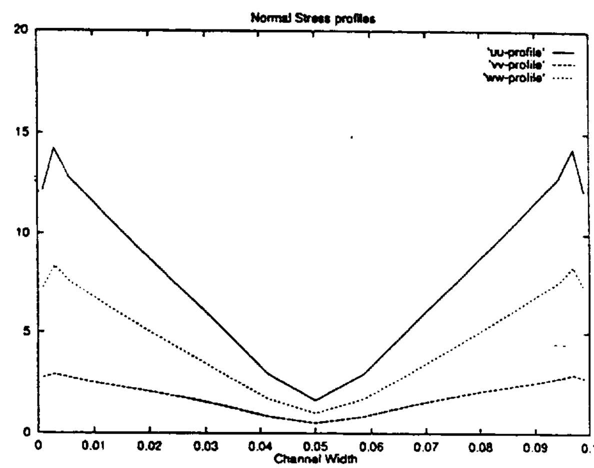


Figure 4. Turbulent intensity profiles across the channel

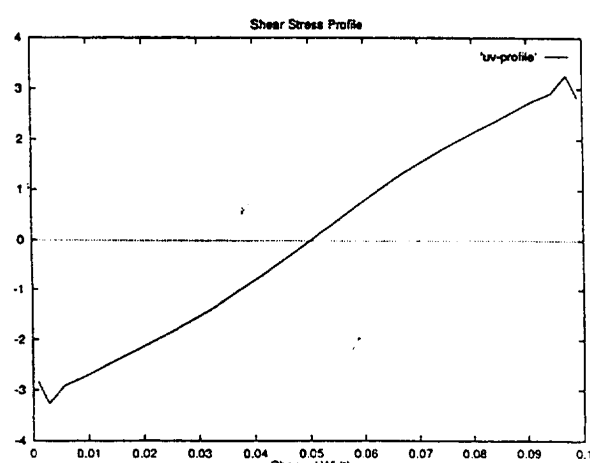


Figure 5. Shear stress profile across the channel

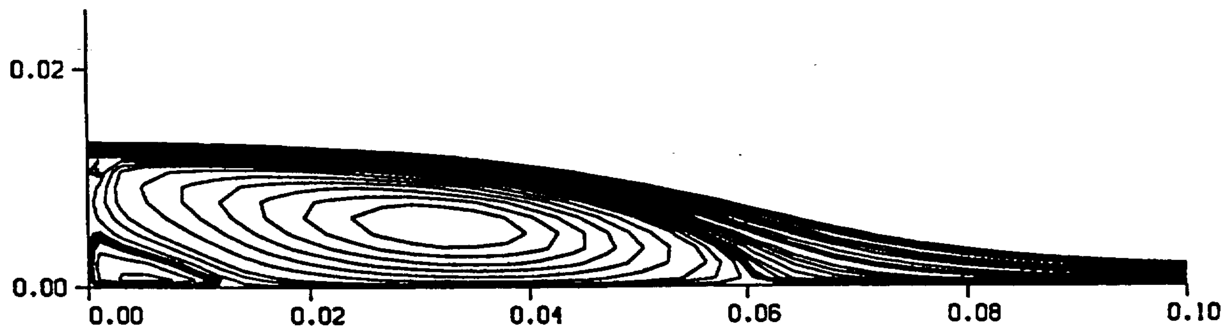


Figure 6. Stream-function contours for the backward facing step flow

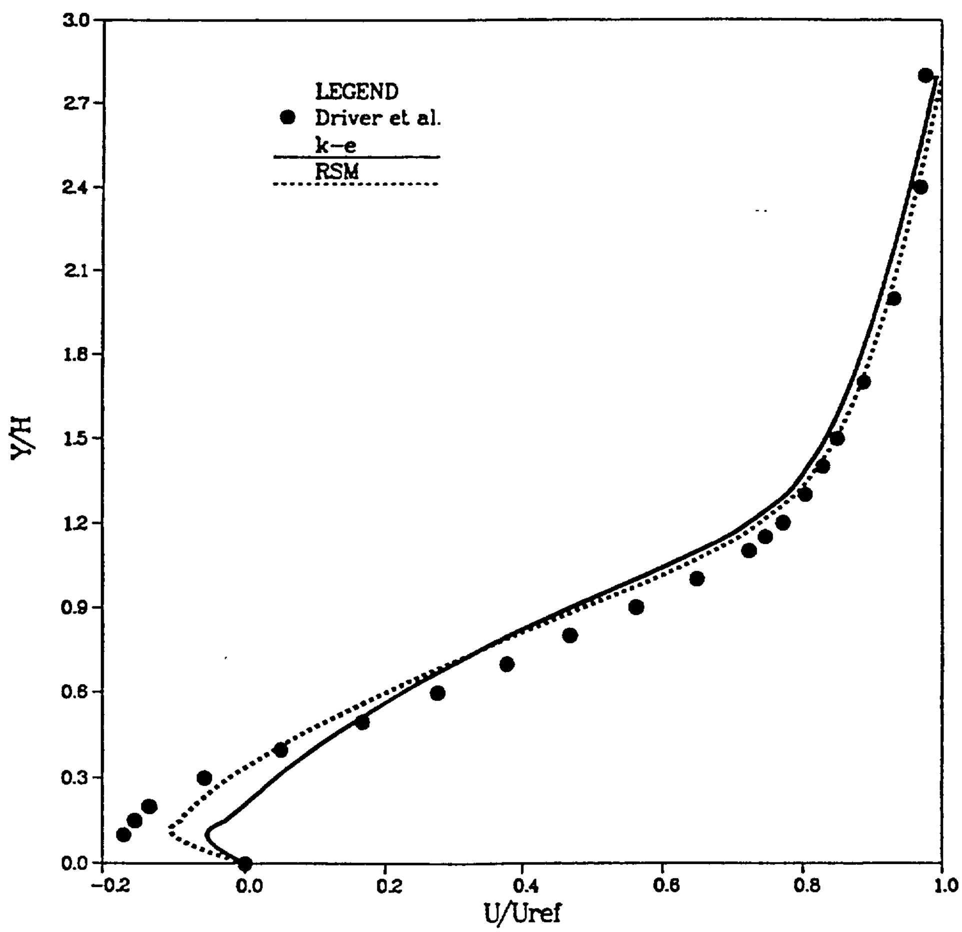


Figure 7. Axial mean velocity profile at *X/H* = *4*

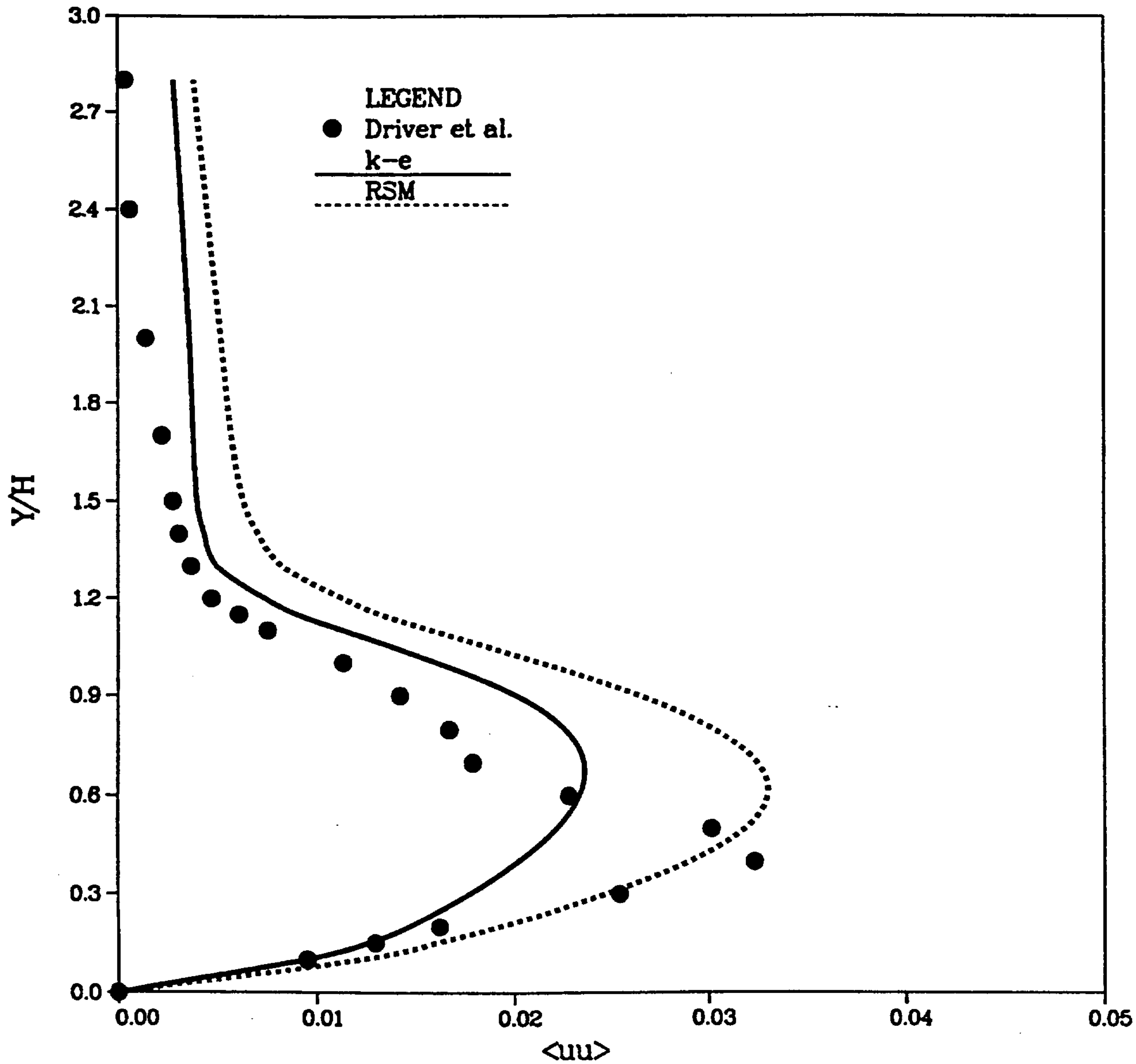


Figure 8. Turbulent intensity $\overline{uu}$ profile (normalized with U_{ref}^2) at $X/H = 4$

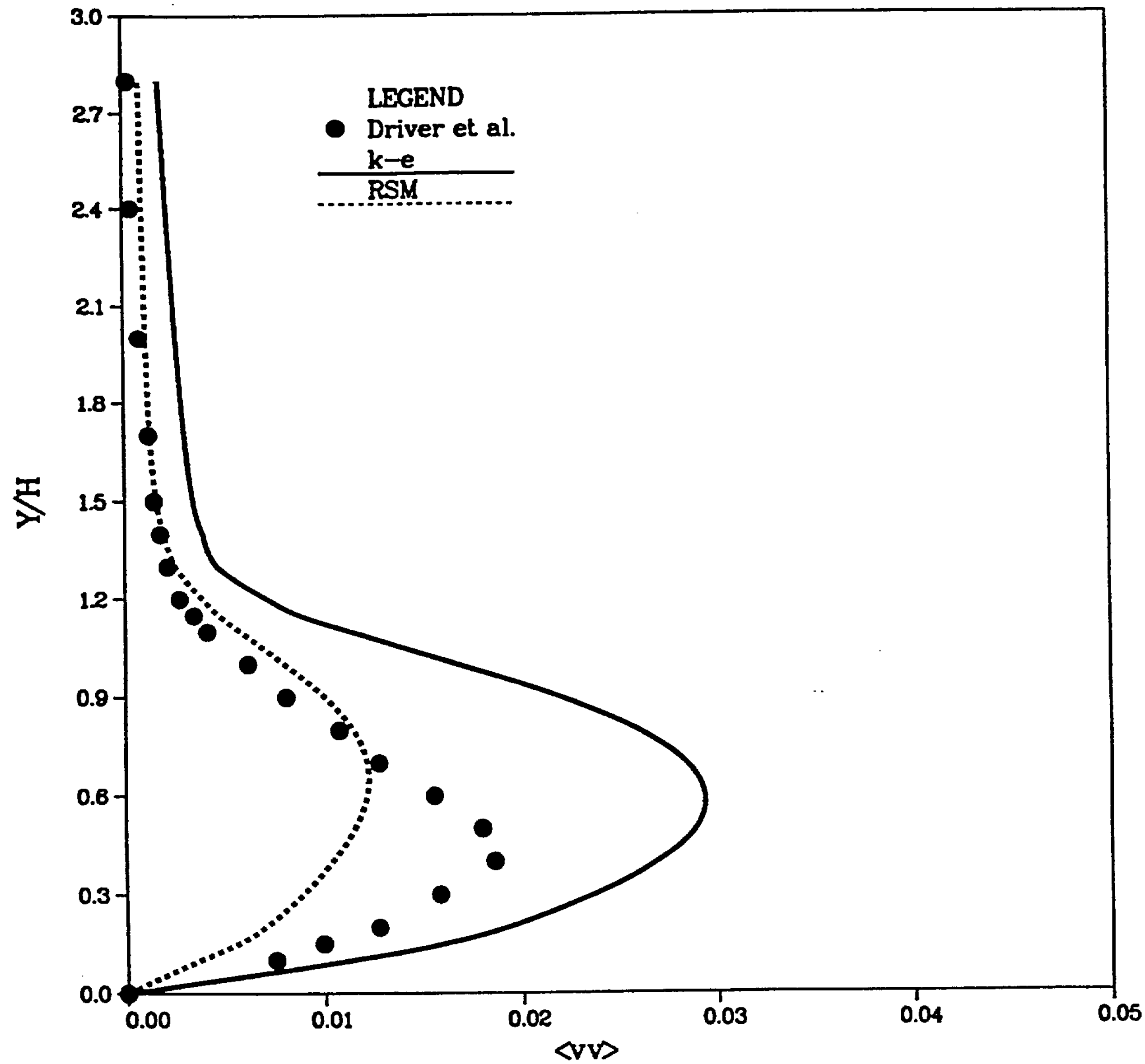


Figure 9. Turbulent stress $\overline{vv}$ profile (normalized with U_{ref}^2) at $X/H = 4$

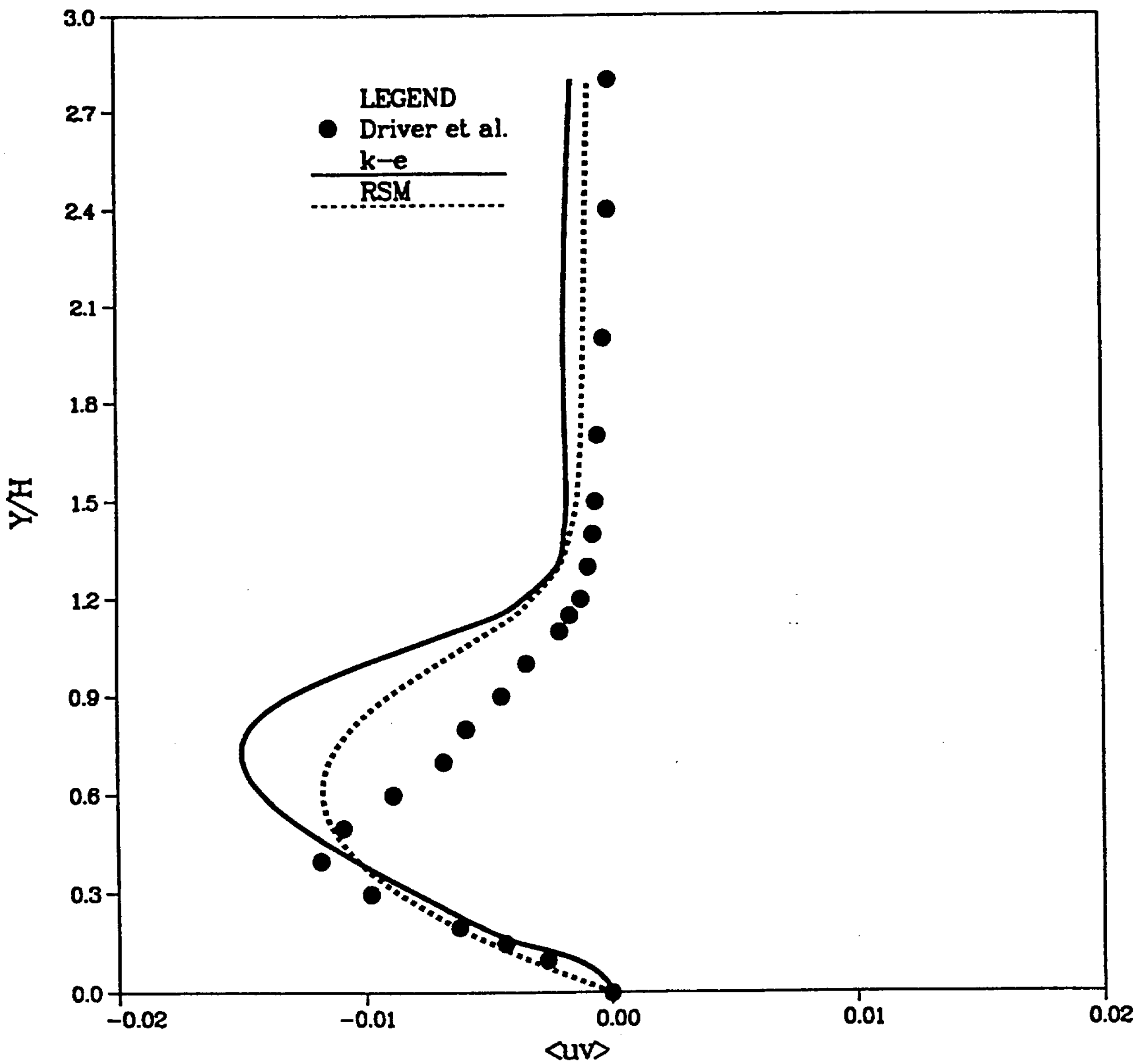


Figure 10. Turbulent shear stress $\overline{uv}$ profile (normalized with U_{ref}^2) at $X/H = 4$

APPENDIX D

2D/Axisymmetric Reynolds Stress Module Deck

The 2D/axisymmetric Reynolds stress module is a FORTRAN source code to solve 2D/Axisymmetric turbulent flow using the full Reynolds stress model based on Launder, Reece and Rodi[1] when interfaced with a main flow solver. The module consists of the main routine RSMOD that calls a number of subroutines to perform different functions that will be explained below.

3.1 Subroutine RSMOD

This is basically the main routine that reads through its argument list different variables from the calling flow solver which are described below.

List of Argument Variable Names

X	Grid node locations in the x or ξ-direction, dimensioned to X(NX*NY)
Y	Grid node locations in the y or η-direction, dimensioned to Y(NX*NY)
FX	Interpolation factor in the x or ξ-direction.
FY	Interpolation factor in the y or η-direction.
ARE	Control cell areas
VOL	Control cell volumes.
R	Radial distance in the axisymmetric geometry or 1. for planar geometry.
DNS	Normal distance of a cell from the south-boundary dimensioned to NX.
DNN	Normal distance of a cell from the north-boundary dimensioned to NX.
DNE	Normal distance of a cell from the east-boundary dimensioned to NY.
DNW	Normal distance of a cell from the west-boundary dimensioned to NY.
U	Axial or ξ-direction velocity, dimensioned to NX*NY.
V	Radial or ψ-direction velocity, dimensioned to NX*NY.
W	Tangential or azimuthal velocity, dimensioned to NX*NY.
TE	Turbulent kinetic energy, dimensioned to NX*NY.
ED	Turbulent energy dissipation rate, dimensioned to NX*NY.
DEN	Density (assumed constant for incompressible flows).
F1	Mass flux at cell faces in the x or ξ-direction, dimensioned to NX*NY.
F2	Mass flux at cell faces in the y or η-direction, dimensioned to NX*NY.

VISCOUS	Laminar viscosity.
VIS	Eddy viscosity, dimensioned to NX*NY.
RESOR	Residual error for the equations solver, dimensioned to 8.
ITBS	Boundary condition flag along the south boundary dimensioned to NX and must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for a wall boundary condition along the south boundary set ITBS to NX*4. Similarly for the other boundaries.
ITBN	Boundary condition flag along the north boundary, dimensioned to NX.
JTBE	Boundary condition flag along the east-boundary dimensioned to NY.
JTBW	Boundary condition flag along the west-boundary dimensioned to NY.
ITER	Iteration number.
ICAL	= 1 for swirl velocity calculations, 0 otherwise.
AKSI	= 1 for axisymmetric flow, 0 otherwise.
RESTART	= 1 if calculations are restarted from a previous run, 0 otherwise.

RSMOD starts by reading the turbulent flow constants, under-relaxation factors and Prandtl/Schmidt numbers for the k and ε equations. These are;

CD1, CD2	constants in the k and ε-equations and are usually set to 1.44 and 1.92 respectively.
CMU, ELOG and CAPPA	also constants in the k and ε-equations and are usually set to 0.09, 9.8 and 0.42 respectively.
GKE	is set to 1 for second-order upwinding of the convective terms in the transport equations.
ALFAKE	is the iteration parameter used in the k and ε-equation solver.
URFVIS	is the underrelaxation factor of the viscosity near the wall.
SORKE(1-8)	are the degree of accuracy for the k, ε, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$, $\overline{vw}$, and $\overline{uw}$-equations solver respectively.
URFKE(1-8)	are the underrelaxation factors for the k, ε, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$, $\overline{vw}$, and $\overline{uw}$-equations respectively.
PRTKE(1-8)	are ratio of Prandtl to Schmidt numbers used in the k, ε, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$, $\overline{vw}$, and $\overline{uw}$ -equations respectively.
C_1, C_2	are constants in the RSM model.

C_{1P} and C_{2P}	are the two constants in the wall-reflection terms of the pressure-strain redistribution term.
C_k and C_ε	constants in the diffusion term of the k and ε -equations.
CUU, CVV, CWW, CUV, CVW, CUW	are the constants multiplying the kinetic energy for the stress values near the wall.
WREFON	= 1 if the wall reflection terms of the pressure-strain term are to be included, = 0 otherwise.

All variable dimensions considered are one-dimensional. The position of any node is defined as IJ = (I,J) = (I-1)*NJ +J, where NI and NJ are the number of grid nodes in the X and Y-directions respectively. It is assumed that grid related data such as cell areas, volumes and interpolation factors be passed to the module from an external grid generator.

Subroutine CALPIJ

This subroutine calculates the production terms of the individual stress components.

Subroutine CALUIUJ

This subroutine solves the transport equations for the turbulent energy (IPHI=1), energy dissipation.(IPHI=2) and Reunolds stresses (IPHI=3, 4, 5, 6, 7, 8 for $\overline{u^2}, \overline{v^2}, \overline{w^2}$, $\overline{uv}$, $\overline{vw}$, and $\overline{uw}$). Daly and Harlow [3] gradient stress diffusion form is used in the module instead of the simplified isotropic diffusivity form. This subroutine calls MODUIUJ subroutine that sets the appropriate boundary conditions for the Reynolds stresses. The set of algebraic difference equations are then solved using Stone's strongly implicit solver SOLSIP.

Subroutine MODPIJ

This subroutine modifies the production terms near the wall using the near wall region model.

Subroutine MODUIUJ

This subroutine calculates the near wall boundary conditions for all the variables.

Subroutine SOLSIP

This subroutine solves the system of linear algebraic equations for all the variables using Stone's Implicit Procedure.

Subroutine WALREF

This subroutine calculates the wall reflection terms in the pressure-strain redistribution correlation. It calculates the wall unit normal vectors and the normal distance away from the wall. This is needed to resolve the wall tangential and normal velocity components that are needed to obtain the near-wall values of the Reynolds stresses.

SUBROUTINE WALPARA

This subroutine calculates the normal and tangential wall unit vectors.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
C     2D/AXISYMMETRIC REYNOLDS STRESS TURBULENCE MODULE                 C
C                                                                       C
C     Rocketdyne CFD Technology Center                                  C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C---REYNOLDS STRESS MODULE -- "RSMODULE"
C
C-------------------------------------------------------------------
      SUBROUTINE RSMOD (X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,
     &                  U,V,W,TE,ED,DEN,F1,F2,VISCOS,VIS,
     &                  RESOR,ITBS,ITBN,JTBE,JTBW,ITER,
     &                  ICAL,AKSI,RESTART)
C--------------------------------------------------------------------
C
      INCLUDE 'gridparam.h'
      INCLUDE 'rsm.h'
C
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),
     &          ARE(NXNY),VOL(NXNY),R(NXNY)
      DIMENSION DNS(NX),DNN(NX),DNE(NY),DNW(NY)
      DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)
      DIMENSION TE(NXNY),ED(NXNY),VIS(NXNY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY),
     &          F1(NXNY),F2(NXNY)
      DIMENSION PHI(NXNY),RESOR(2),TERS(NXNY)
C
C
      NI=NIM+1
      NJ=NJM+1
      NINJ=NI*NJ
C
      IF(ITER.EQ.1) THEN
C
      REWIND 41
C
      READ(41,*)
      READ(41,*)
      READ(41,*) CD1,CD2,CMU,ELOG,CAPPA
      READ(41,*)
      READ(41,*) LRE,LAY2
      READ(41,*)
      READ(41,*) GKE,ALFAKE,URFVIS
      READ(41,*)
      READ(41,*) SORKE(1),NSWPKE(1),URFKE(1),PRTKE(1)
      READ(41,*)
      READ(41,*) SORKE(2),NSWPKE(2),URFKE(2),PRTKE(2)
      READ(41,*)
      READ(41,*) C1, C2, C1P, C2P, CK, CE
      READ(41,*)
      READ(41,*) CUU, CVV, CWW, CUV, CVW, CUW
      READ(41,*)
      READ(41,*) WREFON
C
C      C1=2.5
C      C2=0.55
C      C1P=0.5
C      C2P=0.3
C      CK=0.22
C      CE=0.18
C      CUU=1.098
C      CVV=0.247
C      CWW=0.655
C      CUV=-0.255
C      CVW=-0.255
C      CUW=-0.255
      END IF
C
```



```
C
C--- CALCULATE WALL REFLECTION TERMS
C
       IF (WREFON.EQ.1.) CALL WALREF(X,Y,TE,ED,ITBS,ITBN,JTBW,JTBE)
C
C--- CALCULATE TURBULENCE PRODUCTION TERMS
C
       CALL CALPIJ (X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,U,V,W,
      &             ITBS,ITBN,JTBE,JTBW,iter)
C
C--- CALCULATE REYNOLDS STRESSES FROM DIFFERENTIAL EQUATIONS
C--- INCLUDING TURBULENT KINETIC ENERGY AND DISSIPATION RATE
C--- LOOP BELOW FROM  1 TO 8 FOR K, EPS, U2, V2, W2, UV, VW, UW
C--- BUT IPHI FROM 1 TO 12 FOR ALL VARIABLES U, V, W, P, K, EPS,
C---          U2, V2, W2, UV, VW, UW.
C
       DO II = 1,8
        IPHI = II + 4
         DO IJ=1,NINJ
          IF(II.EQ.1) PHI(IJ)=TE(IJ)
          IF(II.EQ.2) PHI(IJ)=ED(IJ)
          IF(II.EQ.3) PHI(IJ)=U2(IJ)
          IF(II.EQ.4) PHI(IJ)=V2(IJ)
          IF(II.EQ.5) PHI(IJ)=W2(IJ)
          IF(II.EQ.6) PHI(IJ)=UV(IJ)
          IF(II.EQ.7) PHI(IJ)=VW(IJ)
          IF(II.EQ.8) PHI(IJ)=UW(IJ)
         ENDDO
        IF(ICAL(IPHI)) CALL CALUIUJ(ICAL,IPHI,PHI,AKSI,R,U,V,W,DEN,
      &                             TE,ED,VIS,VISCOS,ITBS,
      &                             ITBN,JTBE,JTBW,WREFON,
      &                             ITER)
       ENDDO
C
C--- CHECK TURBULENT ENERGY CALCULATED (TE) WITH BELOW (TERS)
C
       DO 40 IJ=1,NINJ
       TERS(IJ)=0.5*(U2(IJ)+V2(IJ)+W2(IJ))
 40    CONTINUE
C
       RETURN
       END
C
C-----------------------------------------------------
       SUBROUTINE WALREF (X,Y,TE,ED,ITBS,ITBN,JTBW,JTBE)
C-----------------------------------------------------
       INCLUDE 'gridparam.h'
       INCLUDE 'rsm.h'
       DIMENSION X(NXNY),Y(NXNY),TE(NXNY),ED(NXNY)
       DIMENSION FN1S(NX),FN1N(NX),FN1E(NY),FN1W(NY)
       DIMENSION FN2S(NX),FN2N(NX),FN2E(NY),FN2W(NY)
       DIMENSION FT1S(NX),FT1N(NX),FT1E(NY),FT1W(NY)
       DIMENSION FT2S(NX),FT2N(NX),FT2E(NY),FT2W(NY)
       DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
C
C--- CALCULATE NORMAL AND TANGENTIAL WALL UNIT VECTORS
C
       NI=NIM+1
       NJ=NJM+1
C--- ALONG SOUTH & NORTH WALLS---
C
       DO 10 I=2,NIM
       IF(ITBS(I).EQ.4) THEN
       IJ=IMNJ(I)+1
       DXB=X(IJ)-X(IJ-NJ)
       DYB=Y(IJ)-Y(IJ-NJ)
       FHIP=SQRT(DXB**2+DYB**2)
       FT1S(I)=DXB/FHIP
       FT2S(I)=DYB/FHIP
       FN1S(I)=-DXB/FHIP
       FN2S(I)=DXB/FHIP
```

```
      ENDIF
      IF(ITBN(I).EQ.4) THEN
      IJ=IMNJ(I)+NJ
      DXB=X(IJ)-X(IJ-NJ)
      DYB=Y(IJ)-Y(IJ-NJ)
      FHIP=SQRT(DXB**2+DYB**2)
      FT1N(I)=DXB/FHIP
      FT2N(I)=DYB/FHIP
      FN1N(I)=-DYB/FHIP
      FN2N(I)=DXB/FHIP
      ENDIF
 10   CONTINUE
C
C--- ALONG WEST & EAST BOUNDARIES ---
C
      DO 20 J=2,NJM
      IF(JTBW(J).EQ.4) THEN
      IJ=J
      DXB=X(IJ)-X(IJ-1)
      DYB=Y(IJ)-Y(IJ-1)
      FHIP=SQRT(DXB*82+DYB**2)
      FT1W(J)=DXB/FHIP
      FT2W(J)=DYB/FHIP
      FN1W(J)=-DYB/FHIP
      FN2W(J)=DXB/FHIP
      ENDIF
      IF(JTBE(J).EQ.4) THEN
      IJ=IMNJ(NIM)+J
      DXB=X(IJ)-X(IJ-1)
      DYB=Y(IJ)-Y(IJ-1)
      FHIP=SQRT(DXB**2+DYB**2)
      FT1E(J)=DXB/FHIP
      FT2E(J)=DYB/FHIP
      FN1E(J)=-DYB/FHIP
      FN2E(J)=DXB/FHIP
      ENDIF
 20   CONTINUE
C
      CMU25=SQRT(SQRT(CMU))
      CMU75=CMU25**3
      FCON=CMU75/CAPPA
C
      DO 80 I=2,NIM
      DO 80 J=2,NJM
      IJ=IMNJ(I)+J
      RDISNS=0.0
      RDISNN=0.0
      RDISNE=0.0
      RDISNW=0.0
      TE(IJ)=ABS(TE(IJ))
      COEF=FCON*TE(IJ)**1.5/(ED(IJ)+SMALL)
C---START WITH SOUTH BOUNDARY
      IF(ITBS(I).EQ.4) THEN
      IJW=IMNJ(I)+1
      IMJW=IJW-NJ
      DXB=X(IJW)-X(IMJW)
      DYB=Y(IJW)-Y(IMJW)
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNS=1.0/(DISNS+SMALL)
      ENDIF
C---CHECK NORTH BOUNDARY
      IF(ITBN(I).EQ.4) THEN
      IJW=IMNJ(I)+NJM
      IMJW=IJW-NJ
      DXB=X(IJW)-X(IMJW)
```



```
      DYB=Y(IJW)-Y(IMJW)
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNN=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNN=1.0/(DISNN+SMALL)
      ENDIF
C---ALONG THE WEST BOUNDARY
      IF(JTBW(J).EQ.4) THEN
      IJW=J
      IMJW=IJW-1
      DXB=X(IJW)-X(IMJW)
      DYB=Y(IJW)-Y(IMJW)
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNW=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNW=1.0/(DISNW+SMALL)
      ENDIF
C---CHECK EAST BOUNDARY
      IF(JTBE(J).EQ.4) THEN
      IJW=IMNJ(NIM)+J
      IMJW=IJW-1
      DXB=X(IJW)-X(IMJW)
      DYB=Y(IJW)-Y(IMJW)
      XB=HAF*(X(IJW)+X(IMJW))
      YB=HAF*(Y(IJW)+Y(IMJW))
      XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))
      YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))
      DXBP=XBP-XB
      DYBP=YBP-YB
      DISNE=DELTA(DXB,DYB,DXBP,DYBP)
      RDISNE=1.0/(DISNE+SMALL)
      ENDIF
C
      FUNX(IJ)=COEF*(RDISNS*FN1S(I)**2+RDISNN*FN1N(I)**2+
     &              RDISNE*FN1E(J)**2+RDISNW*FN1W(J)**2)
      FUNY(IJ)=COEF*(RDISNS*FN2S(I)**2+RDISNN*FN2N(I)**2+
     &              RDISNE*FN2E(J)**2+RDISNW*FN2W(J)**2)
      FUNXY(IJ)=COEF*(RDISNS*FN1S(I)*FN2S(I)+
     &  RDISNN*FN1N(I)*FN2N(I)+RDISNE*FN1E(J)*FN2E(J)+
     &  RDISNW*FN1W(J)*FN2W(J))
C
 80   CONTINUE
C
      RETURN
      END
C
C
C------------------------------------------------------------------
      SUBROUTINE CALPIJ (X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,U,V,W,
     &                   ITBS,ITBN,JTBE,JTBW,iter)
C------------------------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'rsm.h'
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY)
      DIMENSION ARE(NXNY),VOL(NXNY),R(NXNY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY)
      DIMENSION U1W(NY),U2W(NY),U3W(NY)
      DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)
C
      DO 10 J=2,NJM
      IJ=J
      IJM=IJ-1
      IPJ=IJ+NJ
```

```
      U1W(J)=U(IJ)
      U2W(J)=V(IJ)
      U3W(J)=W(IJ)
   10 CONTINUE
C
      DO 101 I=2,NIM
      J=1
      IJ=IMNJ(I)+J
      UN=U(IJ)
      VN=V(IJ)
      WN=W(IJ)
      DO 102 J=2,NJM
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      IJP=IJ+1
      IJM=IJ-1
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.-FYN
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
C
      US=UN
      VS=VN
      WS=WN
C
      UN=U(IJ)*FYS+U(IJP)*FYN
      VN=V(IJ)*FYS+V(IJP)*FYN
      WN=W(IJ)*FYS+W(IJP)*FYN
C
      UE=U(IJ)*FXW+U(IPJ)*FXE
      VE=V(IJ)*FXW+V(IPJ)*FXE
      WE=W(IJ)*FXW+W(IPJ)*FXE
C
      DUEW=UE-U1W(J)
      DUNS=UN-US
      DVEW=VE-U2W(J)
      DVNS=VN-VS
      DWEW=WE-U3W(J)
      DWNS=WN-WS
C
      DUDX=(DUEW*DYNS-DUNS*DYEW)/ARE(IJ)
      DUDY=(DUNS*DXEW-DUEW*DXNS)/ARE(IJ)
      DVDX=(DVEW*DYNS-DVNS*DYEW)/ARE(IJ)
      DVDY=(DVNS*DXEW-DVEW*DXNS)/ARE(IJ)
      DWDX=(DWEW*DYNS-DWNS*DYEW)/ARE(IJ)
      DWDY=(DWNS*DXEW-DWEW*DXNS)/ARE(IJ)
C
      P11(IJ)=-2.*DEN(IJ)*(U2(IJ)*DUDX+UV(IJ)*DUDY)
      P22(IJ)=-2.*DEN(IJ)*(UV(IJ)*DVDX+V2(IJ)*DVDY-
     &              VW(IJ)*W(IJ)/RP)
      VDR=0.0
      IF(AKSI) VDR=V(IJ)/RP
      P33(IJ)=-2.*DEN(IJ)*(UW(IJ)*DWDX+VW(IJ)*DWDY+
     &              W2(IJ)*VDR)
      P12(IJ)=-DEN(IJ)*(U2(IJ)*DVDX+V2(IJ)*DUDY-
     &          UW(IJ)*W(IJ)/RP-UV(IJ)*VDR)
      P13(IJ)=-DEN(IJ)*(U2(IJ)*DWDX+UV(IJ)*DWDY+
     &          VW(IJ)*DUDY-UW(IJ)*DVDY)
      P23(IJ)=-DEN(IJ)*(UV(IJ)*DWDX+UW(IJ)*DVDX+
     &          V2(IJ)*DWDY+VW(IJ)*DVDY+
     &          VW(IJ)*VDR-W2(IJ)*W(IJ)/RP)
      GEN(IJ)=0.5*(P11(IJ)+P22(IJ)+P33(IJ))
C
      U1W(J)=UE
      U2W(J)=VE
```



```
      U3W(J)=WE
C
  102 CONTINUE
  101 CONTINUE
C
C--- MODIFY GEN-TERMS CLOSE TO A WALL
C
      CALL MODPIJ
      RETURN
      END
C
C
C-----------------------------------------------------------------------
      SUBROUTINE CALUIUJ(ICAL,IPHI,PHI,AKSI,R,U,V,W,DEN,TE,ED,VIS,
     &                 VISCOS,ITBS,ITBN,JTBE,JTBW,WREFON,iter)
C-----------------------------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'rsm.h'
      DIMENSION PHI(NXNY),FXWW(NY),DW(NY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY),TE(NXNY),ED(NXNY),
     #          DEN(NXNY),R(NXNY),A(6,6),B(6),VIS(NXNY)
      DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)
      DIMENSION F222W(NY),F322W(NY),F522W(NY),F622W(NY),
     &          F233W(NY),F333W(NY),F533W(NY),F633W(NY),
     &          F212W(NY),F312W(NY),F412W(NY),F512W(NY),
     &          F123W(NY),F223W(NY),F323W(NY),F523W(NY),
     &          F113W(NY),F213W(NY),F313W(NY),F413W(NY)
C
      URFPHI=1./URF(IPHI)
C
      IJ=1
      PHINE=PHI(IJ)
      PHINW(IJ)=PHINE
C
      DO 11 J=2,NJM
      IJ=J
      IJM=IJ-1
      IPJ=IJ+NJ
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      DXE=XX(IJ)-XX(IJM)
      DYE=YY(IJ)-YY(IJM)
      DXKS=QTR*(XX(IPJ)+XX(IPJ-1)-XX(IJ)-XX(IJM))
      DYKS=QTR*(YY(IPJ)+YY(IPJ-1)-YY(IJ)-YY(IJM))
C
      GAMEU2=0.0
      GAMEV2=0.0
      GAMEUV=0.0
      CKK=CK
      IF(IPHI.EQ.IED) CKK=CE
      IF(ED(IJ).NE.0.0) THEN
      TERM=HAF*TEED(IJ)*CKK*DEN(IJ)*(R(IJ)+R(IJM))
      GAMEU2=TERM*U2(IJ)/AREE
      GAMEV2=TERM*V2(IJ)/AREE
      GAMEUV=TERM*UV(IJ)/AREE
      ENDIF
      DW(IJ)=GAMEU2*DYE**2+GAMEV2*DXE**2
      PHISE=PHINE
      PHINE=PHI(IJ+1)*FY(IJ)+PHI(IJ)*(1.-FY(IJ))
      PHINW(J)=PHINE
      SNSW(J)=0.0
      FXWW(J)=1.0
      IF(JTBW(J).EQ.3.OR.JTBW(J).EQ.4) GO TO 10
      SNSW(J)=-((GAMEU2*DYKS*DYE+GAMEV2*DXKS*DXE)*(PHINE-PHISE)+
     &   2.*GAMEUV*DXE*DYE*(PHI(IPJ)-PHI(IJ))-
     &   GAMEUV*(DXKS*DYE+DYKS*DXE)*(PHINE-PHISE))
   10 CONTINUE
C
C--- ADD GRADIENT TERMS IN RIJ
C
      RP=R(IJ)
C--- V2-EQUATION
```

```
      IF(IPHI.EQ.IV2.AND.ICAL(IRW)) THEN
      F222W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2
      F322W(IJ)=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*VW(IJ)/RP
      F522W(IJ)=VW(IJ)
      F622W(IJ)=VW(IJ)
      ENDIF
C--- W2-EQUATION
      IF(IPHI.EQ.IW2.AND.ICAL(IRW)) THEN
      F233W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2
      F333W(IJ)=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*VW(IJ)/RP
      F533W(IJ)=VW(IJ)
      F633W(IJ)=VW(IJ)
      ENDIF
C--- UV-EQUATION
      IF(IPHI.EQ.IUV.AND.ICAL(IRW)) THEN
      F212W(IJ)=DEN(IJ)*CK*TEED(IJ)*UW(IJ)**2/RP
      F312W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UW(IJ)
      F412W(IJ)=UW(IJ)
      F512W(IJ)=UW(IJ)
      ENDIF
C--- VW-EQUATION
      IF(IPHI.EQ.IVW) THEN
      F123W(IJ)=V2(IJ)-W2(IJ)
      F223W(IJ)=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*(V2(IJ)-W2(IJ))/RP
      F323W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*(V2(IJ)-W2(IJ))
      F523W(IJ)=V2(IJ)-W2(IJ)
      ENDIF
C--- UW-EQUATION
      IF(IPHI.EQ.IUW) THEN
      F113W(IJ)=UV(IJ)
      F213W(IJ)=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*UV(IJ)/RP
      F313W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UV(IJ)
      F413W(IJ)=UV(IJ)
      ENDIF
C
11    CONTINUE
C
      DO 100 I=2,NIM
      J=1
      IJ=IMNJ(I)+J
      IMJ=IJ-NJ
      IJP=IJ+1
      FXE=FX(IJ)
      FXW=1.-FXE
      AREN=HAF*(ARE(IJ)+ARE(IJP))
      DXN=XX(IJ)-XX(IMJ)
      DYN=YY(IJ)-YY(IMJ)
      DXET=QTR*(XX(IJP)+XX(IJP-NJ)-XX(IJ)-XX(IMJ))
      DYET=QTR*(YY(IJP)+YY(IJP-NJ)-YY(IJ)-YY(IMJ))
      GAMNU2=0.0
      GAMNV2=0.0
      GAMNUV=0.0
      CKK=CK
      IF(IPHI.EQ.IED) CKK=CE
      IF(ED(IJ).NE.0.0) THEN
      TERM=HAF*TEED(IJ)*CKK*DEN(IJ)*(R(IJ)+R(IMJ))
      GAMNU2=TERM*U2(IJ)/AREN
      GAMNV2=TERM*V2(IJ)/AREN
      GAMNUV=TERM*UV(IJ)/AREN
      ENDIF
      DN=GAMNU2*DYN**2+GAMNV2*DXN**2
      FYSS=1.0
      PHINE=PHI(IJ+NJ)*FXE+PHI(IJ)*FXW
      SEWN=0.0
      IF(ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 110
      SEWN=-((GAMNU2*DYN*DYET+GAMNV2*DXN*DXET)*(PHINE-PHINW(J))+
     &  2.*GAMNUV*DXN*DYN*(PHI(IJP)-PHI(IJ))-
     &  GAMNUV*(DXET*DYN+DXN*DYET)*(PHINE-PHINW(J)))
110   CONTINUE
      PHINW(J)=PHINE
C
```



```
C--- GRADIENT TERMS IN RIJ
      RP=HAF*(R(IJ)+R(IMJ))
C--- V2-EQUATION
      IF(IPHI.EQ.IV2.AND.ICAL(IRW)) THEN
      F222N=0.0
      F322N=0.0
      F522N=0.0
      F622N=0.0
      IF(RP.GT.0.) THEN
      F222N=DEN(IJ)*TEED(IJ)*CK*VW(IJ)**2
      F322N=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*VW(IJ)/RP
      F522N=VW(IJ)
      F622N=VW(IJ)
      ENDIF
      ENDIF
C--- W2-EQUATION
      IF(IPHI.EQ.IW2.AND.ICAL(IRW)) THEN
      F233N=0.0
      F333N=0.0
      F533N=0.0
      F633N=0.0
      IF(RP.GT.0.) THEN
      F233N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2
      F333N=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*VW(IJ)/RP
      F533N=VW(IJ)
      F633N=VW(IJ)
      ENDIF
      ENDIF
C--- UV-EQUATION
      IF(IPHI.EQ.IUV.AND.ICAL(IRW)) THEN
      F212N=0.0
      F312N=0.0
      F412N=0.0
      F512N=0.0
      IF(RP.GT.0.) THEN
      F212N=DEN(IJ)*CK*TEED(IJ)*UW(IJ)**2/RP
      F312N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UW(IJ)
      F412N=UW(IJ)
      F512N=UW(IJ)
      ENDIF
      ENDIF
C--- VW-EQUATION
      IF(IPHI.EQ.IVW) THEN
      F123N=0.0
      F223N=0.0
      F323N=0.0
      F523N=0.0
      IF(RP.GT.0.) THEN
      F123N=V2(IJ)-W2(IJ)
      F223N=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*(V2(IJ)-W2(IJ))/RP
      F323N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*(V2(IJ)-W2(IJ))
      F523N=V2(IJ)-W2(IJ)
      ENDIF
      ENDIF
C--- UW-EQUATION
      IF(IPHI.EQ.IUW) THEN
      F113N=0.0
      F213N=0.0
      F313N=0.0
      F413N=0.0
      IF(RP.GT.0.) THEN
      F113N=UV(IJ)
      F213N=DEN(IJ)*CK*TEED(IJ)*UW(IJ)*UV(IJ)/RP
      F313N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UV(IJ)
      F413N=UV(IJ)
      ENDIF
      ENDIF
C
C--- THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES
C
      DO 101 J=2,NJM
```

```
      IJ=IMNJ(I)+J
      IPJ=IJ+NJ
      IMJ=IJ-NJ
      IJP=IJ+1
      IJM=IJ-1
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.-FYN
C
      DXE=XX(IJ)-XX(IJM)
      DYE=YY(IJ)-YY(IJM)
      DXN=XX(IJ)-XX(IMJ)
      DYN=YY(IJ)-YY(IMJ)
      DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))
      DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))
      DXET=QTR*(XX(IJP)-XX(IJM)+XX(IJP-NJ)-XX(IJM-NJ))
      DYET=QTR*(YY(IJP)-YY(IJM)+YY(IJP-NJ)-YY(IJM-NJ))
      AREE=HAF*(ARE(IJ)+ARE(IPJ))
      AREN=HAF*(ARE(IJ)+ARE(IJP))
C
      CKK=CK
      IF(IPHI.EQ.IED) CKK=CE
      GAMEU2=HAF*CKK/AREE*(TEED(IJ)*U2(IJ)*DEN(IJ)*FXW+
     &   TEED(IPJ)*U2(IPJ)*DEN(IPJ)*FXE)*(R(IJ)+R(IJM))
      GAMEV2=HAF*CKK/AREE*(TEED(IJ)*V2(IJ)*DEN(IJ)*FXW+
     &   TEED(IPJ)*V2(IPJ)*DEN(IPJ)*FXE)*(R(IJ)+R(IJM))
      GAMEUV=HAF*CKK/AREE*(TEED(IJ)*UV(IJ)*DEN(IJ)*FXW+
     &   TEED(IPJ)*UV(IPJ)*DEN(IPJ)*FXE)*(R(IJ)+R(IJM))
C
      GAMNU2=HAF*CKK/AREN*(TEED(IJ)*U2(IJ)*DEN(IJ)*FYS+
     &   TEED(IJP)*U2(IJP)*DEN(IJP)*FYN)*(R(IJ)+R(IMJ))
      GAMNV2=HAF*CKK/AREN*(TEED(IJ)*V2(IJ)*DEN(IJ)*FYS+
     &   TEED(IJP)*V2(IJP)*DEN(IJP)*FYN)*(R(IJ)+R(IMJ))
      GAMNUV=HAF*CKK/AREN*(TEED(IJ)*UV(IJ)*DEN(IJ)*FYS+
     &   TEED(IJP)*UV(IJP)*DEN(IJP)*FYN)*(R(IJ)+R(IMJ))
C
      DS=DN
      DE=GAMEU2*DYE**2+GAMEV2*DXE**2
      DN=GAMNU2*DYN**2+GAMNV2*DXN**2
C
C--- LINEAR UPWIND DIFFERENCING
C
      AEE=MIN(F1(IJ),0.0)*FX(IPJ)*GRSM
      AWW=-MAX(F1(IMJ),0.0)*(1.0-FXWW(J))*GRSM
      AE1=-MIN(F1(IMJ),0.0)*FXE*GRSM
      AW1=MAX(F1(IJ),0.0)*(1.0-FX(IMJ))*GRSM
      ANN=MIN(F2(IJ),0.0)*FY(IJP)*GRSM
      ASS=-MAX(F2(IJM),0.0)*(1.0-FYSS)*GRSM
      AN1=-MIN(F2(IJM),0.0)*FYN*GRSM
      AS1=MAX(F2(IJ),0.0)*(1.0-FY(IJM))*GRSM
C
      AW(IJ)=DW(J)+MAX(F1(IMJ),0.0)-AWW
      AE(IJ)=DE-MIN(F1(IJ),0.0)-AEE
      AS(IJ)=DS+MAX(F2(IJM),0.0)-ASS
      AN(IJ)=DN-MIN(F2(IJ),0.0)-ANN
C
      DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))
      DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))
      DXET=QTR*(XX(IJP)-XX(IJM)+XX(IJP-NJ)-XX(IJM-NJ))
      DYET=QTR*(YY(IJP)-YY(IJM)+YY(IJP-NJ)-YY(IJM-NJ))
C
      PHISE=PHINE
      PHINE=(PHI(IJ)*FYS+PHI(IJP)*FYN)*FXW+
     &      (PHI(IPJ)*FYS+PHI(IPJ+1)*FYN)*FXE
      SEWS=SEWN
      SEWN=-((GAMNU2*DYN*DYET+GAMNV2*DXN*DXET)*
     &      (PHINE-PHINW(J))-GAMNUV*(DXET*DYN+DXN*DYET)*
     &      (PHINE-PHINW(J))+2.*GAMNUV*DXN*DYN*
     &      (PHI(IJP)-PHI(IJ)))
C
```



```
      SNSE=-((GAMEU2*DYKS*DYE+GAMEV2*DXKS*DXE)*
     &      (PHINE-PHISE)+2.*GAMEUV*DXE*DYE*
     &      (PHI(IPJ)-PHI(IJ))-GAMEUV*(DXKS*DYE+DYKS*DXE)*
     &      (PHINE-PHISE))
C
      IF(I.EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.0
      IF(J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.0
C
      IMJ1=IMJ-NJ
C      IMJJ=MAX0(1,IMJ1)
      IMJJ=MAX(1,IMJ1)
      APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS
      APV(IJ)=APV(IJ)+AEE*PHI(IPJ+NJ)+AWW*PHI(IMJJ)+
     &        ANN*PHI(IJP+1)+ASS*PHI(IJM-1)+
     &        AE1*PHI(IPJ)+AW1*PHI(IMJ)+AN1*PHI(IJP)+
     &        AS1*PHI(IJM)
      APU(IJ)=AEE+AWW+ANN+ASS+AE1+AW1+AN1+AS1
C
C--- SOURCE TERMS FOR THE KINETIC ENERGY EQUATION ----
C
      IF(IPHI.EQ.ITE) THEN
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
      BP(IJ)=APU(IJ)+VOL(IJ)*ED(IJ)*DEN(IJ)/(TE(IJ)+SMALL)
      ENDIF
C
C--- SOURCE TERM OF THE DISSIPATION EQUATION
C
      IF(IPHI.EQ.IED) THEN
      SU(IJ)=APV(IJ)+VOL(IJ)*CD1*ED(IJ)*ABS(GEN(IJ))/
     &                (TE(IJ)+SMALL)
      BP(IJ)=APU(IJ)+VOL(IJ)*CD2*DEN(IJ)*ED(IJ)/
     &                (TE(IJ)+SMALL)
      ENDIF
C
C
C--- ADD SOURCE TERMS DUE TO PRODUCTION, DISSIPATION AND
C--- REDISTRIBUTION TERMS IN THE INDIVIDUAL STRESS EQUATIONS
C
      TH=2./3.
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      RPN=QTR*(R(IJP)+R(IMJ+1)+R(IJ)+R(IMJ))
      RPE=QTR*(R(IPJ)+R(IJ)+R(IPJ-1)+R(IJM))
      DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
C
      FEDK=DEN(IJ)*ED(IJ)/(TE(IJ)+SMALL)
      FKDE=DEN(IJ)*CK*TEED(IJ)
      FKDEN=DEN(IJP)*CK*TEED(IJP)
      FKDEE=DEN(IPJ)*CK*TEED(IPJ)
C
C-------- U2-EQUATION SOURCES -----
C
      IF(IPHI.EQ.IU2) THEN
      SU1=(1.-C2)*P11(IJ)+2./3.*C2*GEN(IJ)+
     &     C1*DEN(IJ)*ED(IJ)*2./3.
      BP1=-(2./3.*DEN(IJ)*ED(IJ)+C1*FEDK*U2(IJ))/
     &     (U2(IJ)+SMALL)
      SU2=2.*C2*C2P*(P11(IJ)-2./3.*GEN(IJ))*FUNX(IJ)+
     &    C1P*FEDK*(V2(IJ)*FUNY(IJ)-UV(IJ)*FUNXY(IJ))-
     &    C2*C2P*((P22(IJ)-2./3.*GEN(IJ))*FUNY(IJ)-
     &    P12(IJ)*FUNXY(IJ))
      BP2=-2.*C1P*FEDK*FUNX(IJ)
C
      SU(IJ)=APV(IJ)+(SU1+SU2)*VOL(IJ)
      BP(IJ)=APU(IJ)-(BP1+BP2)*VOL(IJ)
      ENDIF
C
C-------- V2-EQUATION SOURCES -----
C
```

```
      IF(IPHI.EQ.IV2) THEN
      F222S=F222N
      F322S=F322N
      F522S=F522N
      F622S=F622N
C
      FDUMP=FKDE*VW(IJ)**2
      FDUMN=FKDEN*VW(IJP)**2
      F222N=FDUMP*FYS+FDUMN*FYN
      FDUMP=FKDE*UW(IJ)*VW(IJ)/RP
      FDUMN=FKDEN*UW(IJP)*VW(IJP)/RPN
      F322N=FDUMP*FYS+FDUMN*FYN
      F522N=VW(IJ)*FYS+VW(IJP)*FYN
      F622N=VW(IJ)*FYS+VW(IJP)*FYN
C
      FDUMP=FKDE*VW(IJ)**2
      FDUME=FKDEE*VW(IPJ)**2
      F222E=FDUMP*FXW+FDUME*FXE
      FDUMP=FKDE*UW(IJ)*VW(IJ)/RP
      FDUME=FKDEE*UW(IPJ)*VW(IPJ)/RPE
      F322E=FDUMP*FXW+FDUME*FXE
      F522E=VW(IJ)*FXW+VW(IPJ)*FXE
      F622E=VW(IJ)*FXW+VW(IPJ)*FXE
C
      F2NS=F222N-F222S
      F2EW=F222E-F222W(J)
      F3NS=F322N-F322S
      F3EW=F322E-F322W(J)
      F5NS=F522N-F522S
      F5EW=F522E-F522W(J)
      F6NS=F622N-F622S
      F6EW=F622E-F622W(J)
C
      DF2DY=(F2NS*DXEW-F2EW*DXNS)/ARE(IJ)
      DF3DX=(F3EW*DYNS-F3NS*DYEW)/ARE(IJ)
      DF5DX=(F5EW*DYNS-F5NS*DYEW)/ARE(IJ)
      DF6DY=(F6NS*DXEW-F6EW*DXNS)/ARE(IJ)
C
      SU1=(1.-C2)*P22(IJ)+TH*C2*GEN(IJ)+
     &  TH*DEN(IJ)*C1*ED(IJ)+2.*DEN(IJ)*VW(IJ)*W(IJ)/RP
      BP1=-(TH*DEN(IJ)*ED(IJ)+C1*FEDK*V2(IJ))/(V2(IJ)+SMALL)
      SU2=(C1P*FEDK*U2(IJ) -
     &  C2*C2P*(P11(IJ)-TH*GEN(IJ)))*FUNX(IJ) +
     &  2.*C2*C2P*(P22(IJ)-TH*GEN(IJ))*FUNY(IJ)+
     &  (C2*C2P*P12(IJ)-C1P*FEDK*UV(IJ))*FUNXY(IJ)
      BP2=-2.*C1P*FEDK*FUNY(IJ)
C
      TERM1=0.0
      TERM2=0.0
      TERM3=0.0
      TERM5=0.0
      TERM6=0.0
      BTERM4=0.0
      IF(AKSI) THEN
      TERM1=2.*FKDE*W2(IJ)**2/RP**2
      BTERM4=-2.*FKDE*W2(IJ)/RP**2
      IF(ICAL(IRW)) THEN
      TERM6=-2.*FKDE*VW(IJ)*DF6DY/RP
      TERM5=-2.*FKDE*UW(IJ)*DF5DX/RP
      TERM2=-2.*DF2DY/RP
      TERM3=-2.*DF3DX
      ENDIF
      ENDIF
      R22=TERM1+TERM2+TERM3+TERM5+TERM6
C
      SU(IJ)=APV(IJ)+(SU1+SU2+R22)*VOL(IJ)
      BP(IJ)=APU(IJ)-(BP1+BP2+BTERM4)*VOL(IJ)
C
      ENDIF
C
C--------- W2-EQUATION SOURCES ----------
```



```
      IF(IPHI.EQ.IW2) THEN
      F233S=F233N
      F333S=F333N
      F533S=F533N
      F633S=F633N
C
      F233N=FKDE*VW(IJ)**2*FYS+FKDEN*VW(IJP)**2*FYN
      F333N=FKDE*UW(IJ)*VW(IJ)/RP*FYS +
     &       FKDEN*UW(IJP)*VW(IJP)/RPN*FYN
      F533N=VW(IJ)*FYS+VW(IJP)*FYN
      F633N=VW(IJ)*FYS+VW(IJP)*FYN
C
      F233E=FKDE*VW(IJ)**2*FXW+FKDEE*VW(IPJ)**2*FXE
      F333E=FKDE*UW(IJ)*VW(IJ)/RP*FXW +
     &       FKDEE*UW(IPJ)*VW(IPJ)/RPE*FXE
      F533E=VW(IJ)*FXW+VW(IPJ)*FXE
      F633E=VW(IJ)*FXW+VW(IPJ)*FXE
C
      F2NS=F233N-F233S
      F2EW=F233E-F233W(J)
      F3NS=F333N-F333S
      F3EW=F333E-F333W(J)
      F5NS=F533N-F533S
      F5EW=F533E-F533W(J)
      F6NS=F633N-F633S
      F6EW=F633E-F633W(J)
C
      DF2DY=(F2NS*DXEW-F2EW*DXNS)/ARE(IJ)
      DF3DX=(F3EW*DYNS-F3NS*DYEW)/ARE(IJ)
      DF5DX=(F5EW*DYNS-F5NS*DYEW)/ARE(IJ)
      DF6DY=(F6NS*DXEW-F6EW*DXNS)/ARE(IJ)
C
      SU1=(1.-C2)*P33(IJ)+TH*C2*GEN(IJ)+
     &  TH*DEN(IJ)*C1*ED(IJ)-2.*DEN(IJ)*VW(IJ)*W(IJ)/RP
      BP1=-(TH*DEN(IJ)*ED(IJ)+FEDK*C1*W2(IJ))/(W2(IJ)+SMALL)
      SU2=(C1P*FEDK*U2(IJ)-C2*C2P*
     &      (P11(IJ)-TH*GEN(IJ)))*FUNX(IJ)+(C1P*FEDK*V2(IJ)-
     &       C2*C2P*(P22(IJ)-TH*GEN(IJ)))*FUNY(IJ)+
     &      (2.*C1P*FEDK*UV(IJ)-2.*C2*C2P*P12(IJ))*FUNXY(IJ)
C
      TERM1=0.0
      TERM2=0.0
      TERM3=0.0
      TERM5=0.0
      TERM6=0.0
      BTERM4=0.0
      IF(AKSI) THEN
      TERM1=2.*FKDE*W2(IJ)*V2(IJ)/RP**2
      BTERM4=-2.*FKDE*W2(IJ)/RP**2
      IF(ICAL(IRW)) THEN
      TERM2=2.*DF2DY/RP
      TERM3=2.*DF3DX
      TERM5=2.*FKDE*UW(IJ)*DF5DX/RP
      TERM6=2.*FKDE*VW(IJ)*DF6DY/RP
      ENDIF
      ENDIF
      R33=TERM1+TERM2+TERM3+TERM5+TERM6
      SU(IJ)=APV(IJ)+(SU1+SU2+R33)*VOL(IJ)
      BP(IJ)=APU(IJ)-(BP1+BTERM4)*VOL(IJ)
C
      ENDIF
C
C---------- UV-EQUATION SOURCES ----------
      IF(IPHI.EQ.IUV) THEN
      F212S=F212N
      F312S=F312N
      F412S=F412N
      F512S=F512N
C
      F212N=FKDE*UW(IJ)**2/RP*FYS+FKDEN*UW(IJP)**2/RPN*FYN
      F312N=FKDE*VW(IJ)*UW(IJ)*FYS+FKDEN*VW(IJP)*UW(IJP)*FYN
```

```
      F412N=UW(IJ)*FYS+UW(IJP)*FYN
      F512N=UW(IJ)*FYS+UW(IJP)*FYN
C
      F212E=FKDE*UW(IJ)**2/RP*FXW+FKDEE*UW(IPJ)**2/RPE*FXE
      F312E=FKDE*VW(IJ)*UW(IJ)*FXW+FKDEE*VW(IPJ)*UW(IPJ)*FXE
      F412E=UW(IJ)*FXW+UW(IPJ)*FXE
      F512E=UW(IJ)*FXW+UW(IPJ)*FXE
C
      F2EW=F212E-F212W(J)
      F2NS=F212N-F212S
      F3NS=F312N-F312S
      F3EW=F312E-F312W(J)
      F4EW=F412E-F412W(J)
      F4NS=F412N-F412S
      F5EW=F512E-F512W(J)
      F5NS=F512N-F512S
C
      DF2DX=(F2EW*DYNS-F2NS*DYEW)/ARE(IJ)
      DF3DY=(F3NS*DXEW-F3EW*DXNS)/ARE(IJ)
      DF4DX=(F4EW*DYNS-F4NS*DYEW)/ARE(IJ)
      DF5DY=(F5NS*DXEW-F5EW*DXNS)/ARE(IJ)
C
      SU1=(1.-C2)*P12(IJ)+1.5*C2*C2P*P12(IJ)*
     &  (FUNX(IJ)+FUNY(IJ))+1.5*C2*C2P*
     &  (P11(IJ)+P22(IJ)-4./3.*GEN(IJ))*FUNXY(IJ)-
     &  1.5*C1P*FEDK*(U2(IJ)+V2(IJ))*FUNXY(IJ)
      BP1=-(C1*FEDK+1.5*C1P*FEDK*
     &     (FUNX(IJ)+FUNY(IJ)))
C
      BTERM1=0.0
      TERM2=0.0
      TERM3=0.0
      TERM4=0.0
      TERM5=0.0
      IF(AKSI) THEN
      BTERM1=-FKDE*W2(IJ)/RP**2
      IF(ICAL(IRW)) THEN
      TERM5=DEN(IJ)*UW(IJ)*W(IJ)/RP
      TERM2=-DF2DX
      TERM3=-DF3DY/RP
      TERM4=-FKDE/RP*(UW(IJ)*DF4DX+VW(IJ)*DF5DY)
      ENDIF
      ENDIF
      R12=TERM2+TERM3+TERM4+TERM5
C
      SU(IJ)=APV(IJ)+(SU1+R12)*VOL(IJ)
      BP(IJ)=APU(IJ)-(BP1+BTERM1)*VOL(IJ)
C
      ENDIF
C
C---------- VW-EQUATION SOURCES ----------
      IF(IPHI.EQ.IVW) THEN
C
      F123S=F123N
      F223S=F223N
      F323S=F323N
      F523S=F523N
C
      F123N=(V2(IJ)-W2(IJ))*FYS+(V2(IJP)-W2(IJP))*FYN
      F223N=FKDE*UW(IJ)*(V2(IJ)-W2(IJ))/RP*FYS+
     &      FKDEN*UW(IJP)*(V2(IJP)-W2(IJP))/RPN*FYN
      F323N=FKDE*VW(IJ)*(V2(IJ)-W2(IJ))*FYS+
     &      FKDEN*VW(IJP)*(V2(IJP)-W2(IJP))*FYN
      F523N=(V2(IJ)-W2(IJ))*FYS+(V2(IJP)-W2(IJP))*FYN
C
      F123E=(V2(IJ)-W2(IJ))*FXW+(V2(IPJ)-W2(IPJ))*FXE
      F223E=FKDE*UW(IJ)*(V2(IJ)-W2(IJ))/RP*FXW+
     &      FKDEE*UW(IPJ)*(V2(IPJ)-W2(IPJ))/RPE*FXE
      F323E=FKDE*VW(IJ)*(V2(IJ)-W2(IJ))*FXW+
     &      FKDEE*VW(IPJ)*(V2(IPJ)-W2(IPJ))*FXE
      F523E=(V2(IJ)-W2(IJ))*FXW+(V2(IPJ)-W2(IPJ))*FXE
```



```
C
      F1NS=F123N-F123S
      F1EW=F123E-F123W(J)
      F2EW=F223E-F223W(J)
      F2NS=F223N-F223S
      F3NS=F323N-F323S
      F3EW=F323E-F323W(J)
      F5EW=F523E-F523W(J)
      F5NS=F523N-F523S
C
      DF1DY=(F1NS*DXEW-F1EW*DXNS)/ARE(IJ)
      DF2DX=(F2EW*DYNS-F2NS*DYEW)/ARE(IJ)
      DF3DY=(F3NS*DXEW-F3EW*DXNS)/ARE(IJ)
      DF5DX=(F5EW*DYNS-F5NS*DYEW)/ARE(IJ)
C
      SU1=(1.-C2)*P23(IJ)+1.5*C2*C2P*(P23(IJ)*FUNY(IJ)+
     & P13(IJ)*FUNXY(IJ))-1.5*C1P*FEDK*UW(IJ)*FUNXY(IJ)
      BP1=-(C1*FEDK+1.5*C1P*FEDK*FUNY(IJ))
      BTERM1=FKDE*DF1DY/RP
      BTERM4=-4.*FKDE*W2(IJ)/RP**2
      TERM2=DF2DX
      TERM3=DF3DY/RP
      TERM5=FKDE*UW(IJ)*DF5DX/RP
      TERM6=-DEN(IJ)*(V2(IJ)-W2(IJ))*W(IJ)/RP
      R23=TERM2+TERM3+TERM5+TERM6
C
      SU(IJ)=APV(IJ)+(SU1+R23)*VOL(IJ)
      BP(IJ)=APU(IJ)-(BP1+BTERM1+BTERM4)*VOL(IJ)
C
      ENDIF
C
C---------- UW-EQUATION SOURCES ----------
      IF(IPHI.EQ.IUW) THEN
C
      F113S=F113N
      F213S=F213N
      F313S=F313N
      F413S=F413N
C
      F113N=UV(IJ)*FYS+UV(IJP)*FYN
      F213N=FKDE*UW(IJ)*UV(IJ)/RP*FYS+
     &      FKDEN*UW(IJP)*UV(IJP)/RPN*FYN
      F313N=FKDE*VW(IJ)*UV(IJ)*FYS+
     &      FKDEN*VW(IJP)*UV(IJP)*FYN
      F413N=UV(IJ)*FYS+UV(IJP)*FYN
C
      F113E=UV(IJ)*FXW+UV(IPJ)*FXE
      F213E=FKDE*UW(IJ)*UV(IJ)/RP*FXW+
     &      FKDEE*UW(IPJ)*UV(IPJ)/RPE*FXE
      F313E=FKDE*VW(IJ)*UV(IJ)*FXW+
     &      FKDEE*VW(IPJ)*UV(IPJ)*FXE
      F413E=UV(IJ)*FXW+UV(IPJ)*FXE
C
      F1NS=F113N-F113S
      F1EW=F113E-F113W(J)
      F2EW=F213E-F213W(J)
      F2NS=F213N-F213S
      F3NS=F313N-F313S
      F3EW=F313E-F313W(J)
      F4EW=F413E-F413W(J)
      F4NS=F413N-F413S
C
      DF1DY=(F1NS*DXEW-F1EW*DXNS)/ARE(IJ)
      DF2DX=(F2EW*DYNS-F2NS*DYEW)/ARE(IJ)
      DF3DY=(F3NS*DXEW-F3EW*DXNS)/ARE(IJ)
      DF4DX=(F4EW*DYNS-F4NS*DYEW)/ARE(IJ)
C
      SU1=(1.-C2)*P13(IJ)+1.5*C2*C2P*(P13(IJ)*FUNX(IJ)+
     & P23(IJ)*FUNXY(IJ))-1.5*C1P*FEDK*VW(IJ)*FUNXY(IJ)
      BP1=-(C1*FEDK+1.5*C1P*FEDK*FUNX(IJ))
      TERM1=FKDE*VW(IJ)*DF1DY/RP
```

```
      TERM2=DF2DX
      TERM3=DF3DY/RP
      BTERM4=FKDE*DF4DX/RP
      BTERM5=-FKDE*W2(IJ)/RP**2
      TERM6=-DEN(IJ)*UV(IJ)*W(IJ)/RP
      R13=TERM1+TERM2+TERM3+TERM6
C
      SU(IJ)=APV(IJ)+(SU1+R13)*VOL(IJ)
      BP(IJ)=APU(IJ)-(BP1+BTERM4+BTERM5)*VOL(IJ)
C
      ENDIF
C
      F222W(J)=F222E
      F322W(J)=F322E
      F522W(J)=F522E
      F622W(J)=F622E
C
      F233W(J)=F233E
      F333W(J)=F333E
      F533W(J)=F533E
      F633W(J)=F633E
C
      F212W(J)=F212E
      F312W(J)=F312E
      F412W(J)=F412E
      F512W(J)=F512E
C
      F123W(J)=F123E
      F223W(J)=F223E
      F323W(J)=F323E
      F523W(J)=F523E
C
      F113W(J)=F113E
      F213W(J)=F213E
      F313W(J)=F313E
      F413W(J)=F413E
C
      SNSW(J)=SNSE
      PHINW(J)=PHINE
      FYSS=FY(IJM)
      FXWW(J)=FX(IMJ)
      DW(J)=DE
C
  101 CONTINUE
  100 CONTINUE
C
C----- PROBLEM MODIFICATION AND BOUNDARY CONDITIONS
C
      CALL MODUIUJ
C
      DO 200 I=2,NIM
      DO 201 J=2,NJM
      IJ=IMNJ(I)+J
      IMJ=IJ-NJ
      IJM=IJ-1
      AP(IJ)=AW(IJ)+AE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)
      AP(IJ)=AP(IJ)*URFPHI
      SU(IJ)=SU(IJ)+(1.-URF(IPHI))*AP(IJ)*PHI(IJ)
      IF(IPHI.EQ.IU2) APUU(IJ)=1./AP(IJ)
      IF(IPHI.EQ.IV2) APVV(IJ)=1./AP(IJ)
      IF(IPHI.EQ.IUV) APUV(IJ)=1./AP(IJ)
      IF(IPHI.EQ.IVW) APVW(IJ)=1./AP(IJ)
      IF(IPHI.EQ.IUW) APUW(IJ)=1./AP(IJ)
  201 CONTINUE
  200 CONTINUE
C
C---- SOLVING F.D. EQUATIONS
C
      CALL SIPSOL(PHI,IPHI)
C
       RETURN
```



```
       END
C
C
C**********************************************************************
      SUBROUTINE CALDUVW
C**********************************************************************
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C---SUBROUTINE TO CALCULATE MEAN-VELOCITY GRADIENTS AND
C---THE REYNOLDS STRESS TERMS AT THE CELL FACES
C---TO INCREASE THE COUPLING BETWEEN THE MEAN-VELOCITY AND
C---REYNOLDS STRESSES IN THE MOMENTUM EQUATIONS.
C
      INCLUDE 'rsm.common.block'
      COMMON/DU123/ DU2EW(NXNY),DU2NS(NXNY),DUVNS(NXNY),
     &  DUVEW(NXNY),DV2EW(NXNY),DV2NS(NXNY),DUWEW(NXNY),
     &  DUWNS(NXNY),DVWEW(NXNY),DVWNS(NXNY)
C
      DIMENSION DU1(NXNY),DU2(NXNY),DV1(NXNY),DV2(NXNY),
     &          DW1(NXNY),DW2(NXNY)
      DIMENSION U1W(NY),V1W(NY),W1W(NY)
      DIMENSION UVW(NY),UUW(NY),VVW(NY),UWW(NY),VWW(NY)
C
      DO 302 J=2,NJM
      I=1
      IJ=IMNJ(I)+J
      U1W(J)=U(IJ)
      V1W(J)=V(IJ)
      W1W(J)=W(IJ)
      UVW(J)=UV(IJ)*R(IJ)*DEN(IJ)
      UUW(J)=U2(IJ)*R(IJ)*DEN(IJ)
      VVW(J)=V2(IJ)*R(IJ)*DEN(IJ)
      UWW(J)=UW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
      VWW(J)=VW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
  302 CONTINUE
C
      DO 300 I=2,NIM
      IJ=IMNJ(I)+1
      U1N=U(IJ)
      V1N=V(IJ)
      W1N=W(IJ)
      DO 301 J=2,NJM
      IJ=IMNJ(I)+J
      U1S=U1N
      V1S=V1N
      W1S=W1N
      U1E=U(IJ+NJ)*FX(IJ)+U(IJ)*(1.-FX(IJ))
      V1E=V(IJ+NJ)*FX(IJ)+V(IJ)*(1.-FX(IJ))
      W1E=W(IJ+NJ)*FX(IJ)+W(IJ)*(1.-FX(IJ))
      U1N=U(IJ+1)*FY(IJ)+U(IJ)*(1.-FY(IJ))
      V1N=V(IJ+1)*FY(IJ)+V(IJ)*(1.-FY(IJ))
      W1N=W(IJ+1)*FY(IJ)+W(IJ)*(1.-FY(IJ))
C
      DU1(IJ)=U1E-U1W(J)
      DU2(IJ)=U1N-U1S
      DV1(IJ)=V1E-V1W(J)
      DV2(IJ)=V1N-V1S
      DW1(IJ)=W1E-W1W(J)
      DW2(IJ)=W1N-W1S
C
      U1W(J)=U1E
      V1W(J)=V1E
      W1W(J)=W1E
  301 CONTINUE
  300 CONTINUE
C
C
C---- CALCULATE DU2EW(IJ) APPEARING IN THE U-MOMENTUM
C
      DO 321 I=2,NIM
      IJ=IMNJ(I)+1
      UUN=U2(IJ)*R(IJ)*DEN(IJ)
```

```
      UVN=UV(IJ)*R(IJ)*DEN(IJ)
      VVN=V2(IJ)*R(IJ)*DEN(IJ)
      UWN=UW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
      VWN=VW(IJ)*R(IJ)*R(IJ)*DEN(IJ)
C
      DO 320 J=2,NJM
      IJ=IMNJ(I)+J
      UUS=UUN
      VVS=VVN
      UVS=UVN
      UWS=UWN
      VWS=VWN
      IPJ=IJ+NJ
      IJP=IJ+1
      IJM=IJ-1
      IMJ=IJ-NJ
      FXE=FX(IJ)
      FXW=1.-FXE
      FYN=FY(IJ)
      FYS=1.-FYN
      DXE=XX(IJ)-XX(IJM)
      DYE=YY(IJ)-YY(IJM)
      DXN=XX(IJ)-XX(IMJ)
      DYN=YY(IJ)-YY(IMJ)
      RE=HAF*(R(IJ)+R(IJM))
      RN=HAF*(R(IJ)+R(IMJ))
      DYEP=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
      DYKP=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
      DXKP=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
      DXEP=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
      DYEE=HAF*(YY(IPJ)+YY(IJ)-YY(IPJ-1)-YY(IJM))
      DXEE=HAF*(XX(IPJ)+XX(IJ)-XX(IPJ-1)-XX(IJM))
      DXKN=HAF*(XX(IJP)+XX(IJ)-XX(IJP-NJ)-XX(IMJ))
      DYKN=HAF*(YY(IJP)+YY(IJ)-YY(IJP-NJ)-YY(IMJ))
      RP=QTR*(R(IJ)+R(IJM)+R(IMJ)+R(IMJ-1))
      RPE=QTR*(R(IJ)+R(IJM)+R(IPJ)+R(IPJ-1))
      RPN=QTR*(R(IJ)+R(IMJ)+R(IJP)+R(IMJ+1))
      DYKE=HAF*(YY(IPJ)+YY(IPJ-1)-YY(IJ)-YY(IJM))
      DXKE=HAF*(XX(IPJ)+XX(IPJ-1)-XX(IJ)-XX(IJM))
      DYEN=HAF*(YY(IJP)+YY(IJP-NJ)-YY(IJ)-YY(IMJ))
      DXEN=HAF*(XX(IJP)+XX(IJP-NJ)-XX(IJ)-XX(IMJ))
      XP=QTR*(XX(IJ)+XX(IMJ)+XX(IJM)+XX(IMJ-1))
      YP=QTR*(YY(IJ)+YY(IMJ)+YY(IJM)+YY(IMJ-1))
      XE=QTR*(XX(IPJ)+XX(IJ)+XX(IJM)+XX(IPJ-1))
      YE=QTR*(YY(IPJ)+YY(IJ)+YY(IJM)+YY(IPJ-1))
      XN=QTR*(XX(IJP)+XX(IJP-NJ)+XX(IMJ)+XX(IJ))
      YN=QTR*(YY(IJP)+YY(IJP-NJ)+YY(IMJ)+YY(IJ))
C
      RDENE=R(IJ)*DEN(IJ)*FXW+R(IPJ)*DEN(IPJ)*FXE
      RDENN=R(IJ)*DEN(IJ)*FYS+R(IJP)*DEN(IJP)*FYN
      RRDENE=R(IJ)*R(IJ)*DEN(IJ)*FXW+R(IPJ)*R(IPJ)*DEN(IPJ)*FXE
      RRDENN=R(IJ)*R(IJ)*DEN(IJ)*FYS+R(IJP)*R(IJP)*DEN(IJP)*FYN
      DENE=DEN(IJ)*FXW+DEN(IPJ)*FXE
      DENN=DEN(IJ)*FYS+DEN(IJP)*FYN
C
      D1P=-DYEP*APUU(IJ)*RP
      D2P=DYKP*APUU(IJ)*RP
      E1P=-DXKP*APUU(IJ)*RP
      E2P=DXEP*APUU(IJ)*RP
C
      D1PE=-DYEE*APUU(IPJ)*RPE
      D2PE=DYKE*APUU(IPJ)*RPE
      E1PE=-DXKE*APUU(IPJ)*RPE
      E2PE=DXEE*APUU(IPJ)*RPE
C
      D1PN=-DYEN*APUU(IJP)*RPN
      D2PN=DYKN*APUU(IJP)*RPN
      E1PN=-DXKN*APUU(IJP)*RPN
      E2PN=DXEN*APUU(IJP)*RPN
C
      APUUE=APUU(IJ)*FXW+APUU(IPJ)*FXE
```



```
      D1E=-APUUE*DYE*RE
      D2E=APUUE*(YE-YP)*RE
      E1E=-APUUE*(XE-XP)*RE
      E2E=APUUE*DXE*RE
C
      APUUN=APUU(IJ)*FYS+APUU(IJP)*FYN
      D1N=-APUUN*(YN-YP)*RN
      D2N=APUUN*DYN*RN
      E1N=-APUUN*DXN*RN
      E2N=APUUN*(XN-XP)*RN
C
      UPE=U(IPJ)-U(IJ)
      VPE=V(IPJ)-V(IJ)
      WPE=W(IPJ)-W(IJ)
      UPN=U(IJP)-U(IJ)
      VPN=V(IJP)-V(IJ)
      WPN=W(IJP)-W(IJ)
C
C--- CONSIDER TERMS IN UUE-EQ.----
      AA=1.-2./3.*C2+1./3.*C2*C2P*(FUNY(IJ)+4.*FUNX(IJ))
      AAE=1.-2./3.*C2+1./3.*C2*C2P*(FUNY(IPJ)+4.*FUNX(IPJ))
      AAN=1.-2./3.*C2+1./3.*C2*C2P*(FUNY(IJP)+4.*FUNX(IJP))
      BB=1./3.*C2-2./3.*C2*C2P*(FUNX(IJ)+FUNY(IJ))
      BBE=1./3.*C2-2./3.*C2*C2P*(FUNX(IPJ)+FUNY(IPJ))
      BBN=1./3.*C2-2./3.*C2*C2P*(FUNX(IJP)+FUNY(IJP))
      CC=1./3.*C2*(1.+C2P*(FUNY(IJ)-2.*FUNX(IJ)))
      CCE=1./3.*C2*(1.+C2P*(FUNY(IPJ)-2.*FUNX(IPJ)))
      CCN=1./3.*C2*(1.+C2P*(FUNY(IJP)-2.*FUNX(IJP)))
      DD=C2*C2P*FUNXY(IJ)
      DDE=C2*C2P*FUNXY(IPJ)
      DDN=C2*C2P*FUNXY(IJP)
C
      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
      TA1E=2.*DEN(IPJ)*AAE*U2(IPJ)*D1PE*DU1(IPJ)
      TA2P=2.*DEN(IJ)*AA*UV(IJ)*E2P*DU1(IJ)
      TA2E=2.*DEN(IPJ)*AAE*UV(IPJ)*E2PE*DU1(IPJ)
      TB1P=2.*DEN(IJ)*BB*UV(IJ)*D1P*DV1(IJ)
      TB1E=2.*DEN(IPJ)*BBE*UV(IPJ)*D1PE*DV1(IPJ)
      TB2P=2.*DEN(IJ)*BB*V2(IJ)*E2P*DV1(IJ)
      TB2E=2.*DEN(IPJ)*BBE*V2(IPJ)*E2PE*DV1(IPJ)
      TC1P=2.*DEN(IJ)*CC*UW(IJ)*D1P*DW1(IJ)
      TC1E=2.*DEN(IPJ)*CCE*UW(IPJ)*D1PE*DW1(IPJ)
      TC2P=2.*DEN(IJ)*CC*VW(IJ)*E2P*DW1(IJ)
      TC2E=2.*DEN(IPJ)*CCE*VW(IPJ)*E2PE*DW1(IPJ)
      TD1P=DEN(IJ)*DD*U2(IJ)*D1P*DV1(IJ)
      TD1E=DEN(IPJ)*DDE*U2(IPJ)*D1PE*DV1(IPJ)
      TD2P=DEN(IJ)*DD*V2(IJ)*E2P*DU1(IJ)
      TD2E=DEN(IPJ)*DDE*V2(IPJ)*E2PE*DU1(IPJ)
C
      SA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*FXW +
     &      2.*DEN(IPJ)*AAE*U2(IPJ)*D1PE*FXE
      SA2E=2.*DEN(IJ)*AA*UV(IJ)*E2P*FXW +
     &      2.*DEN(IPJ)*AAE*UV(IPJ)*E2PE*FXE
      SB1E=2.*DEN(IJ)*BB*UV(IJ)*D1P*FXW +
     &      2.*DEN(IPJ)*BBE*UV(IPJ)*D1PE*FXE
      SB2E=2.*DEN(IJ)*BB*V2(IJ)*E2P*FXW +
     &      2.*DEN(IPJ)*BBE*V2(IPJ)*E2PE*FXE
      SC1E=2.*DEN(IJ)*CC*UW(IJ)*D1P*FXW +
     &      2.*DEN(IPJ)*CCE*UW(IPJ)*D1PE*FXE
      SC2E=2.*DEN(IJ)*CC*VW(IJ)*E2P*FXW +
     &      2.*DEN(IPJ)*CCE*VW(IPJ)*E2PE*FXE
      SD1E=DEN(IJ)*DD*U2(IJ)*D1P*FXW +
     &      DEN(IPJ)*DDE*U2(IPJ)*D1PE*FXE
      SD2E=DEN(IJ)*DD*V2(IJ)*E2P*FXW +
     &      DEN(IPJ)*DDE*V2(IPJ)*E2PE*FXE
C
      UUE=(U2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
     &      TC2P-TD1P-TD2P)*FXW +
     &   (U2(IPJ)-TA1E-TA2E-TB1E-TB2E-TC1E-
     &      TC2E-TD1E-TD2E)*FXE +
     &   SA1E*UPE+SA2E*UPE+SB1E*VPE+SB2E*VPE+
```

```
     &   SC1E*WPE+SC2E*WPE+SD1E*VPE+SD2E*UPE
C
      UUE=UUE*RDENE
C
C---CONSIDER TERMS IN UUN-EQ.----
C
      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DU2(IJ)
      TA1N=2.*DEN(IJP)*AAN*U2(IJP)*D2PN*DU2(IJP)
      TA2P=2.*DEN(IJ)*AA*UV(IJ)*E1P*DU2(IJ)
      TA2N=2.*DEN(IJP)*AAN*UV(IJP)*E1PN*DU2(IJP)
      TB1P=2.*DEN(IJ)*BB*UV(IJ)*D2P*DV2(IJ)
      TB1N=2.*DEN(IJP)*BBN*UV(IJP)*D2PN*DV2(IJP)
      TB2P=2.*DEN(IJ)*BB*V2(IJ)*E1P*DV2(IJ)
      TB2N=2.*DEN(IJP)*BBN*V2(IJP)*E1PN*DV2(IJP)
      TC1P=2.*DEN(IJ)*CC*UW(IJ)*D2P*DW2(IJ)
      TC1N=2.*DEN(IJP)*CCN*UW(IJP)*D2PN*DW2(IJP)
      TC2P=2.*DEN(IJ)*CC*VW(IJ)*E1P*DW2(IJ)
      TC2N=2.*DEN(IJP)*CCN*VW(IJP)*E1PN*DW2(IJP)
      TD1P=DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
      TD1N=DEN(IJP)*DDN*U2(IJP)*D2PN*DV2(IJP)
      TD2P=DEN(IJ)*DD*V2(IJ)*E1P*DU2(IJ)
      TD2N=DEN(IJP)*DDN*V2(IJP)*E1PN*DU2(IJP)
C
      SA1N=2.*DEN(IJ)*AA*U2(IJ)*D2P*FYS +
     &      2.*DEN(IJP)*AAN*U2(IJP)*D2PN*FYN
      SA2N=2.*DEN(IJ)*AA*UV(IJ)*E1P*FYS +
     &      2.*DEN(IJP)*AAN*UV(IJP)*E1PN*FYN
      SB1N=2.*DEN(IJ)*BB*UV(IJ)*D2P*FYS +
     &      2.*DEN(IJP)*BBN*UV(IJP)*D2PN*FYN
      SB2N=2.*DEN(IJ)*BB*V2(IJ)*E1P*FYS +
     &      2.*DEN(IJP)*BBN*V2(IJP)*E1PN*FYN
      SC1N=2.*DEN(IJ)*CC*UW(IJ)*D2P*FYS +
     &      2.*DEN(IJP)*CCN*UW(IJP)*D2PN*FYN
      SC2N=2.*DEN(IJ)*CC*VW(IJ)*E1P*FYS +
     &      2.*DEN(IJP)*CCN*VW(IJP)*E1PN*FYN
      SD1N=DEN(IJ)*DD*U2(IJ)*D2P*FYS +
     &      DEN(IJP)*DDN*U2(IJP)*D2PN*FYN
      SD2N=DEN(IJ)*DD*V2(IJ)*E1P*FYS +
     &      DEN(IJP)*DDN*V2(IJP)*E1PN*FYN
C
      UUN=(U2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
     &      TC2P-TD1P-TD2P)*FYS +
     &    (U2(IJP)-TA1N-TA2N-TB1N-TB2N-TC1N-
     &      TC2N-TD1N-TD2N)*FYN +
     &   SA1N*UPN+SA2N*UPN+SB1N*VPN+SB2N*VPN+
     &   SC1N*WPN+SC2N*WPN+SD1N*VPN+SD2N*UPN
C
      UUN=UUN*RDENN
C
C---CONSIDER TERMS IN UVE-EQ.----
C
      D1P=-DYEP*APUV(IJ)*RP
      D2P=DYKP*APUV(IJ)*RP
      E1P=-DXKP*APUV(IJ)*RP
      E2P=DXEP*APUV(IJ)*RP
C
      D1PE=-DYEE*APUV(IPJ)*RPE
      D2PE=DYKE*APUV(IPJ)*RPE
      E1PE=-DXKE*APUV(IPJ)*RPE
      E2PE=DXEE*APUV(IPJ)*RPE
C
      D1PN=-DYEN*APUV(IJP)*RPN
      D2PN=DYKN*APUV(IJP)*RPN
      E1PN=-DXKN*APUV(IJP)*RPN
      E2PN=DXEN*APUV(IJP)*RPN
C
      APUVE=APUV(IJ)*FXW+APUV(IPJ)*FXE
      D1E=-APUVE*DYE*RE
      D2E=APUVE*(YE-YP)*RE
      E1E=-APUVE*(XE-XP)*RE
      E2E=APUVE*DXE*RE
```



```
C
      APUVN=APUV(IJ)*FYS+APUV(IJP)*FYN
      D1N=-APUVN*(YN-YP)*RN
      D2N=APUVN*DYN*RN
      E1N=-APUVN*DXN*RN
      E2N=APUVN*(XN-XP)*RN
C
      AA=HAF*C2*C2P*FUNXY(IJ)
      AAE=HAF*C2*C2P*FUNXY(IPJ)
      AAN=HAF*C2*C2P*FUNXY(IJP)
      BB=AA
      BBE=AAE
      BBN=AAN
      CC=-C2*C2P*FUNXY(IJ)
      CCE=-C2*C2P*FUNXY(IPJ)
      CCN=-C2*C2P*FUNXY(IJP)
      DD=1.-C2+1.5*C2*C2P*(FUNX(IJ)+FUNY(IJ))
      DDE=1.-C2+1.5*C2*C2P*(FUNX(IPJ)+FUNY(IPJ))
      DDN=1.-C2+1.5*C2*C2P*(FUNX(IJP)+FUNY(IJP))
C
      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
      TA1E=2.*DEN(IPJ)*AAE*U2(IPJ)*D1PE*DU1(IPJ)
      TA2P=2.*DEN(IJ)*AA*UV(IJ)*E2P*DU1(IJ)
      TA2E=2.*DEN(IPJ)*AAE*UV(IPJ)*E2PE*DU1(IPJ)
      TB1P=2.*DEN(IJ)*BB*UV(IJ)*D1P*DV1(IJ)
      TB1E=2.*DEN(IPJ)*BBE*UV(IPJ)*D1PE*DV1(IPJ)
      TB2P=2.*DEN(IJ)*BB*V2(IJ)*E2P*DV1(IJ)
      TB2E=2.*DEN(IPJ)*BBE*V2(IPJ)*E2PE*DV1(IPJ)
      TC1P=2.*DEN(IJ)*CC*UW(IJ)*D1P*DW1(IJ)
      TC1E=2.*DEN(IPJ)*CCE*UW(IPJ)*D1PE*DW1(IPJ)
      TC2P=2.*DEN(IJ)*CC*VW(IJ)*E2P*DW1(IJ)
      TC2E=2.*DEN(IPJ)*CCE*VW(IPJ)*E2PE*DW1(IPJ)
      TD1P=DD*DEN(IJ)*U2(IJ)*D1P*DV1(IJ)
      TD1E=DDE*DEN(IPJ)*U2(IPJ)*D1PE*DV1(IPJ)
      TD2P=DD*DEN(IJ)*V2(IJ)*E2P*DU1(IJ)
      TD2E=DDE*DEN(IPJ)*V2(IPJ)*E2PE*DU1(IPJ)
C
      SA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*FXW +
     &      2.*DEN(IPJ)*AAE*U2(IPJ)*D1PE*FXE
      SA2E=2.*DEN(IJ)*AA*UV(IJ)*E2P*FXW +
     &      2.*DEN(IPJ)*AAE*UV(IPJ)*E2PE*FXE
      SB1E=2.*DEN(IJ)*BB*UV(IJ)*D1P*FXW +
     &      2.*DEN(IPJ)*BBE*UV(IPJ)*D1PE*FXE
      SB2E=2.*DEN(IJ)*BB*V2(IJ)*E2P*FXW +
     &      2.*DEN(IPJ)*BBE*V2(IPJ)*E2PE*FXE
      SC1E=2.*DEN(IJ)*CC*UW(IJ)*D1P*FXW +
     &      2.*DEN(IPJ)*CCE*UW(IPJ)*D1PE*FXE
      SC2E=2.*DEN(IJ)*CC*VW(IJ)*E2P*FXW +
     &      2.*DEN(IPJ)*CCE*VW(IPJ)*E2PE*FXE
      SD1E=DD*DEN(IJ)*U2(IJ)*D1P*FXW +
     &      DDE*DEN(IPJ)*U2(IPJ)*D1PE*FXE
      SD2E=DD*DEN(IJ)*V2(IJ)*E2P*FXW +
     &      DDE*DEN(IPJ)*V2(IPJ)*E2PE*FXE
C
      UVE=(UV(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
     &      TC2P-TD1P-TD2P)*FXW +
     &    (UV(IPJ)-TA1E-TA2E-TB1E-TB2E-TC1E-
     &      TC2E-TD1E-TD2E)*FXE +
     &   SA1E*UPE+SA2E*UPE+SB1E*VPE+SB2E*VPE+
     &   SC1E*WPE+SC2E*WPE+SD1E*VPE+SD2E*UPE
C
      UVE=UVE*RDENE
C
C---CONSIDER TERMS IN UVN-EQ.----
C
      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DU2(IJ)
      TA1N=2.*DEN(IJP)*AAN*U2(IJP)*D2PN*DU2(IJP)
      TA2P=2.*DEN(IJ)*AA*UV(IJ)*E1P*DU2(IJ)
      TA2N=2.*DEN(IJP)*AAN*UV(IJP)*E1PN*DU2(IJP)
      TB1P=2.*DEN(IJ)*BB*UV(IJ)*D2P*DV2(IJ)
      TB1N=2.*DEN(IJP)*BBN*UV(IJP)*D2PN*DV2(IJP)
```

```
      TB2P=2.*DEN(IJ)*BB*V2(IJ)*E1P*DV2(IJ)
      TB2N=2.*DEN(IJP)*BBN*V2(IJP)*E1PN*DV2(IJP)
      TC1P=2.*DEN(IJ)*CC*UW(IJ)*D2P*DW2(IJ)
      TC1N=2.*DEN(IJP)*CCN*UW(IJP)*D2PN*DW2(IJP)
      TC2P=2.*DEN(IJ)*CC*VW(IJ)*E1P*DW2(IJ)
      TC2N=2.*DEN(IJP)*CCN*VW(IJP)*E1PN*DW2(IJP)
      TD1P=DD*DEN(IJ)*U2(IJ)*D2P*DV2(IJ)
      TD1N=DDN*DEN(IJP)*U2(IJP)*D2PN*DV2(IJP)
      TD2P=DD*DEN(IJ)*V2(IJ)*E1P*DU2(IJ)
      TD2N=DDN*DEN(IJP)*V2(IJP)*E1PN*DU2(IJP)
C
      SA1N=2.*DEN(IJ)*AA*U2(IJ)*D2P*FYS +
     &     2.*DEN(IJP)*AAN*U2(IJP)*D2PN*FYN
      SA2N=2.*DEN(IJ)*AA*UV(IJ)*E1P*FYS +
     &     2.*DEN(IJP)*AAN*UV(IJP)*E1PN*FYN
      SB1N=2.*DEN(IJ)*BB*UV(IJ)*D2P*FYS +
     &     2.*DEN(IJP)*BBN*UV(IJP)*D2PN*FYN
      SB2N=2.*DEN(IJ)*BB*V2(IJ)*E1P*FYS +
     &     2.*DEN(IJP)*BBN*V2(IJP)*E1PN*FYN
      SC1N=2.*DEN(IJ)*CC*UW(IJ)*D2P*FYS +
     &     2.*DEN(IJP)*CCN*UW(IJP)*D2PN*FYN
      SC2N=2.*DEN(IJ)*CC*VW(IJ)*E1P*FYS +
     &     2.*DEN(IJP)*CCN*VW(IJP)*E1PN*FYN
      SD1N=DD*DEN(IJ)*U2(IJ)*D2P*FYS +
     &     DDN*DEN(IJP)*U2(IJP)*D2PN*FYN
      SD2N=DD*DEN(IJ)*V2(IJ)*E1P*FYS +
     &     DDN*DEN(IJP)*V2(IJP)*E1PN*FYN
C
      UVN=(UV(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
     &     TC2P-TD1P-TD2P)*FYS +
     &   (UV(IJP)-TA1N-TA2N-TB1N-TB2N-TC1N-
     &     TC2N-TD1N-TD2N)*FYN +
     &  SA1N*UPN+SA2N*UPN+SB1N*VPN+SB2N*VPN+
     &  SC1N*WPN+SC2N*WPN+SD1N*VPN+SD2N*UPN
C
      UVN=UVN*RDENN
C
C
C---CONSIDER TERMS IN VVE-EQ.----
C
      D1P=-DYEP*APVV(IJ)*RP
      D2P=DYKP*APVV(IJ)*RP
      E1P=-DXKP*APVV(IJ)*RP
      E2P=DXEP*APVV(IJ)*RP
C
      D1PE=-DYEE*APVV(IPJ)*RPE
      D2PE=DYKE*APVV(IPJ)*RPE
      E1PE=-DXKE*APVV(IPJ)*RPE
      E2PE=DXEE*APVV(IPJ)*RPE
C
      D1PN=-DYEN*APVV(IJP)*RPN
      D2PN=DYKN*APVV(IJP)*RPN
      E1PN=-DXKN*APVV(IJP)*RPN
      E2PN=DXEN*APVV(IJP)*RPN
C
      APVVE=APVV(IJ)*FXW+APVV(IPJ)*FXE
      D1E=-APVVE*DYE*RE
      D2E=APVVE*(YE-YP)*RE
      E1E=-APVVE*(XE-XP)*RE
      E2E=APVVE*DXE*RE
C
      APVVN=APVV(IJ)*FYS+APVV(IJP)*FYN
      D1N=-APVVN*(YN-YP)*RN
      D2N=APVVN*DYN*RN
      E1N=-APVVN*DXN*RN
      E2N=APVVN*(XN-XP)*RN
C
      AA=1./3.*C2-2./3.*C2*C2P*(FUNX(IJ)+FUNY(IJ))
      AAE=1./3.*C2-2./3.*C2*C2P*(FUNX(IPJ)+FUNY(IPJ))
      AAN=1./3.*C2-2./3.*C2*C2P*(FUNX(IJP)+FUNY(IJP))
      BB=1.-2./3.*C2+1./3.*C2*C2P*(4.*FUNY(IJ)+FUNX(IJ))
```



```
      BBE=1.-2./3.*C2+1./3.*C2*C2P*
     &    (4.*FUNY(IPJ)+FUNX(IPJ))
      BBN=1.-2./3.*C2+1./3.*C2*C2P*
     &    (4.*FUNY(IJP)+FUNX(IJP))
      CC=1./3.*C2+1./3.*C2*C2P*(FUNX(IJ)-2.*FUNY(IJ))
      CCE=1./3.*C2+1./3.*C2*C2P*(FUNX(IPJ)-2.*FUNY(IPJ))
      CCN=1./3.*C2+1./3.*C2*C2P*(FUNX(IJP)-2.*FUNY(IJP))
      DD=C2*C2P*FUNXY(IJ)
      DDE=C2*C2P*FUNXY(IPJ)
      DDN=C2*C2P*FUNXY(IJP)
C
      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
      TA1E=2.*DEN(IPJ)*AAE*U2(IPJ)*D1PE*DU1(IPJ)
      TA2P=2.*DEN(IJ)*AA*UV(IJ)*E2P*DU1(IJ)
      TA2E=2.*DEN(IPJ)*AAE*UV(IPJ)*E2PE*DU1(IPJ)
      TB1P=2.*DEN(IJ)*BB*UV(IJ)*D1P*DV1(IJ)
      TB1E=2.*DEN(IPJ)*BBE*UV(IPJ)*D1PE*DV1(IPJ)
      TB2P=2.*DEN(IJ)*BB*V2(IJ)*E2P*DV1(IJ)
      TB2E=2.*DEN(IPJ)*BBE*V2(IPJ)*E2PE*DV1(IPJ)
      TC1P=2.*DEN(IJ)*CC*UW(IJ)*D1P*DW1(IJ)
      TC1E=2.*DEN(IPJ)*CCE*UW(IPJ)*D1PE*DW1(IPJ)
      TC2P=2.*DEN(IJ)*CC*VW(IJ)*E2P*DW1(IJ)
      TC2E=2.*DEN(IPJ)*CCE*VW(IPJ)*E2PE*DW1(IPJ)
      TD1P=DEN(IJ)*DD*U2(IJ)*D1P*DV1(IJ)
      TD1E=DEN(IPJ)*DDE*U2(IPJ)*D1PE*DV1(IPJ)
      TD2P=DEN(IJ)*DD*V2(IJ)*E2P*DU1(IJ)
      TD2E=DEN(IPJ)*DDE*V2(IPJ)*E2PE*DU1(IPJ)
C
      SA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*FXW +
     &   2.*DEN(IPJ)*AAE*U2(IPJ)*D1PE*FXE
      SA2E=2.*DEN(IJ)*AA*UV(IJ)*E2P*FXW +
     &   2.*DEN(IPJ)*AAE*UV(IPJ)*E2PE*FXE
      SB1E=2.*DEN(IJ)*BB*UV(IJ)*D1P*FXW +
     &   2.*DEN(IPJ)*BBE*UV(IPJ)*D1PE*FXE
      SB2E=2.*DEN(IJ)*BB*V2(IJ)*E2P*FXW +
     &   2.*DEN(IPJ)*BBE*V2(IPJ)*E2PE*FXE
      SC1E=2.*DEN(IJ)*CC*UW(IJ)*D1P*FXW +
     &   2.*DEN(IPJ)*CCE*UW(IPJ)*D1PE*FXE
      SC2E=2.*DEN(IJ)*CC*VW(IJ)*E2P*FXW +
     &   2.*DEN(IPJ)*CCE*VW(IPJ)*E2PE*FXE
      SD1E=DEN(IJ)*DD*U2(IJ)*D1P*FXW +
     &      DEN(IPJ)*DDE*U2(IPJ)*D1PE*FXE
      SD2E=DEN(IJ)*DD*V2(IJ)*E2P*FXW +
     &      DEN(IPJ)*DDE*V2(IPJ)*E2PE*FXE
C
      VVE=(V2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
     &      TC2P-TD1P-TD2P)*FXW +
     &   (V2(IPJ)-TA1E-TA2E-TB1E-TB2E-TC1E-
     &      TC2E-TD1E-TD2E)*FXE +
     &   SA1E*UPE+SA2E*UPE+SB1E*VPE+SB2E*VPE+
     &   SC1E*WPE+SC2E*WPE+SD1E*VPE+SD2E*UPE
C
      VVE=VVE*RDENE
C
C
C---CONSIDER TERMS IN VVN-EQ.----
C
      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DU2(IJ)
      TA1N=2.*DEN(IJP)*AAN*U2(IJP)*D2PN*DU2(IJP)
      TA2P=2.*DEN(IJ)*AA*UV(IJ)*E1P*DU2(IJ)
      TA2N=2.*DEN(IJP)*AAN*UV(IJP)*E1PN*DU2(IJP)
      TB1P=2.*DEN(IJ)*BB*UV(IJ)*D2P*DV2(IJ)
      TB1N=2.*DEN(IJP)*BBN*UV(IJP)*D2PN*DV2(IJP)
      TB2P=2.*DEN(IJ)*BB*V2(IJ)*E1P*DV2(IJ)
      TB2N=2.*DEN(IJP)*BBN*V2(IJP)*E1PN*DV2(IJP)
      TC1P=2.*DEN(IJ)*CC*UW(IJ)*D2P*DW2(IJ)
      TC1N=2.*DEN(IJP)*CCN*UW(IJP)*D2PN*DW2(IJP)
      TC2P=2.*DEN(IJ)*CC*VW(IJ)*E1P*DW2(IJ)
      TC2N=2.*DEN(IJP)*CCN*VW(IJP)*E1PN*DW2(IJP)
      TD1P=DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
      TD1N=DEN(IJP)*DDN*U2(IJP)*D2PN*DV2(IJP)
```

```
      TD2P=DEN(IJ)*DD*V2(IJ)*E1P*DU2(IJ)
      TD2N=DEN(IJP)*DDN*V2(IJP)*E1PN*DU2(IJP)
C
      SA1N=2.*DEN(IJ)*AA*U2(IJ)*D2P*FYS +
     &      2.*DEN(IJP)*AAN*U2(IJP)*D2PN*FYN
      SA2N=2.*DEN(IJ)*AA*UV(IJ)*E1P*FYS +
     &      2.*DEN(IJP)*AAN*UV(IJP)*E1PN*FYN
      SB1N=2.*DEN(IJ)*BB*UV(IJ)*D2P*FYS +
     &      2.*DEN(IJP)*BBN*UV(IJP)*D2PN*FYN
      SB2N=2.*DEN(IJ)*BB*V2(IJ)*E1P*FYS +
     &      2.*DEN(IJP)*BBN*V2(IJP)*E1PN*FYN
      SC1N=2.*DEN(IJ)*CC*UW(IJ)*D2P*FYS +
     &      2.*DEN(IJP)*CCN*UW(IJP)*D2PN*FYN
      SC2N=2.*DEN(IJ)*CC*VW(IJ)*E1P*FYS +
     &      2.*DEN(IJP)*CCN*VW(IJP)*E1PN*FYN
      SD1N=DEN(IJ)*DD*U2(IJ)*D2P*FYS +
     &      DEN(IJP)*DDN*U2(IJP)*D2PN*FYN
      SD2N=DEN(IJ)*DD*V2(IJ)*E1P*FYS +
     &      DEN(IJP)*DDN*V2(IJP)*E1PN*FYN
C
      VVN=(V2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
     &      TC2P-TD1P-TD2P)*FYS +
     &    (V2(IJP)-TA1N-TA2N-TB1N-TB2N-TC1N-
     &      TC2N-TD1N-TD2N)*FYN +
     & SA1N*UPN+SA2N*UPN+SB1N*VPN+SB2N*VPN+
     & SC1N*WPN+SC2N*WPN+SD1N*VPN+SD2N*UPN
C
      VVN=VVN*RDENN
C
C
C---CONSIDER TERMS IN UWE-EQ.----
C
      D1P=-DYEP*APUW(IJ)*RP
      D2P=DYKP*APUW(IJ)*RP
      E1P=-DXKP*APUW(IJ)*RP
      E2P=DXEP*APUW(IJ)*RP
C
      D1PE=-DYEE*APUW(IPJ)*RPE
      D2PE=DYKE*APUW(IPJ)*RPE
      E1PE=-DXKE*APUW(IPJ)*RPE
      E2PE=DXEE*APUW(IPJ)*RPE
C
      D1PN=-DYEN*APUW(IJP)*RPN
      D2PN=DYKN*APUW(IJP)*RPN
      E1PN=-DXKN*APUW(IJP)*RPN
      E2PN=DXEN*APUW(IJP)*RPN
C
      APUWE=APUW(IJ)*FXW+APUW(IPJ)*FXE
      D1E=-APUWE*DYE*RE
      D2E=APUWE*(YE-YP)*RE
      E1E=-APUWE*(XE-XP)*RE
      E2E=APUWE*DXE*RE
C
      APUWN=APUW(IJ)*FYS+APUW(IJP)*FYN
      D1N=-APUWN*(YN-YP)*RN
      D2N=APUWN*DYN*RN
      E1N=-APUWN*DXN*RN
      E2N=APUWN*(XN-XP)*RN
C
      AA=1.-C2+1.5*C2*C2P*FUNX(IJ)
      AAE=1.-C2+1.5*C2*C2P*FUNX(IPJ)
      AAN=1.-C2+1.5*C2*C2P*FUNX(IJP)
      BB=1.5*C2*C2P*FUNXY(IJ)
      BBE=1.5*C2*C2P*FUNXY(IPJ)
      BBN=1.5*C2*C2P*FUNXY(IJP)
C
      TA1P=AA*DEN(IJ)*U2(IJ)*D1P*DW1(IJ)
      TA1E=AAE*DEN(IPJ)*U2(IPJ)*D1PE*DW1(IPJ)
      TA2P=AA*DEN(IJ)*UV(IJ)*E2P*DW1(IJ)
      TA2E=AAE*DEN(IPJ)*UV(IPJ)*E2PE*DW1(IPJ)
      TA3P=AA*DEN(IJ)*VW(IJ)*E2P*DU1(IJ)
```



```
      TA3E=AAE*DEN(IPJ)*VW(IPJ)*E2PE*DU1(IPJ)
      TA4P=AA*DEN(IJ)*UW(IJ)*E2P*DV1(IJ)
      TA4E=AAE*DEN(IPJ)*UW(IPJ)*E2PE*DV1(IPJ)
      TB1P=BB*DEN(IJ)*UV(IJ)*D1P*DW1(IJ)
      TB1E=BBE*DEN(IPJ)*UV(IPJ)*D1PE*DW1(IPJ)
      TB2P=BB*DEN(IJ)*UW(IJ)*D1P*DV1(IJ)
      TB2E=BBE*DEN(IPJ)*UW(IPJ)*D1PE*DV1(IPJ)
      TB3P=BB*DEN(IJ)*V2(IJ)*E2P*DW1(IJ)
      TB3E=BBE*DEN(IPJ)*V2(IPJ)*E2PE*DW1(IPJ)
      TB4P=BB*DEN(IJ)*VW(IJ)*E2P*DV1(IJ)
      TB4E=BBE*DEN(IPJ)*VW(IPJ)*E2PE*DV1(IPJ)
C
      SA1E=AA*DEN(IJ)*U2(IJ)*D1P*FXW +
     &      AAE*DEN(IPJ)*U2(IPJ)*D1PE*FXE
      SA2E=AA*DEN(IJ)*UV(IJ)*E2P*FXW +
     &      AAE*DEN(IPJ)*UV(IPJ)*E2PE*FXE
      SA3E=AA*DEN(IJ)*VW(IJ)*E2P*FXW +
     &      AAE*DEN(IPJ)*VW(IPJ)*E2PE*FXE
      SA4E=AA*DEN(IJ)*UW(IJ)*E2P*FXW +
     &      AAE*DEN(IPJ)*UW(IPJ)*E2PE*FXE
      SB1E=BB*DEN(IJ)*UV(IJ)*D1P*FXW +
     &      BBE*DEN(IPJ)*UV(IPJ)*D1PE*FXE
      SB2E=BB*DEN(IJ)*UW(IJ)*D1P*FXW +
     &      BBE*DEN(IPJ)*UW(IPJ)*D1PE*FXE
      SB3E=BB*DEN(IJ)*V2(IJ)*E2P*FXW +
     &      BBE*DEN(IPJ)*V2(IPJ)*E2PE*FXE
      SB4E=BB*DEN(IJ)*VW(IJ)*E2P*FXW +
     &      BBE*DEN(IPJ)*VW(IPJ)*E2PE*FXE
C
      UWE=(UW(IJ)-TA1P-TA2P-TA3P+TA4P-
     &      TB1P-TB2P-TB3P-TB4P)*FXW +
     &    (UW(IPJ)-TA1E-TA2E-TA3E+TA4E-
     &      TB1E-TB2E-TB3E-TB4E)*FXE +
     & SA1E*WPE+SA2E*WPE+SA3E*UPE-SA4E*VPE+
     & SB1E*WPE+SB2E*VPE+SB3E*WPE+SB4E*VPE
C
      UWE=UWE*RRDENE
C
C
C---CONSIDER TERMS IN UWN-EQ.----
C
      TA1P=AA*DEN(IJ)*U2(IJ)*D2P*DW2(IJ)
      TA1N=AAN*DEN(IJP)*U2(IJP)*D2PN*DW2(IJP)
      TA2P=AA*DEN(IJ)*UV(IJ)*E1P*DW2(IJ)
      TA2N=AAN*DEN(IJP)*UV(IJP)*E1PN*DW2(IJP)
      TA3P=AA*DEN(IJ)*VW(IJ)*E1P*DU2(IJ)
      TA3N=AAN*DEN(IJP)*VW(IJP)*E1PN*DU2(IJP)
      TA4P=AA*DEN(IJ)*UW(IJ)*E1P*DV2(IJ)
      TA4N=AAN*DEN(IJP)*UW(IJP)*E1PN*DV2(IJP)
      TB1P=BB*DEN(IJ)*UV(IJ)*D2P*DW2(IJ)
      TB1N=BBN*DEN(IJP)*UV(IJP)*D2PN*DW2(IJP)
      TB2P=BB*DEN(IJ)*UW(IJ)*D2P*DV2(IJ)
      TB2N=BBN*DEN(IJP)*UW(IJP)*D2PN*DV2(IJP)
      TB3P=BB*DEN(IJ)*V2(IJ)*E1P*DW2(IJ)
      TB3N=BBN*DEN(IJP)*V2(IJP)*E1PN*DW2(IJP)
      TB4P=BB*DEN(IJ)*VW(IJ)*E1P*DV2(IJ)
      TB4N=BBN*DEN(IJP)*VW(IJP)*E1PN*DV2(IJP)
C
      SA1N=AA*DEN(IJ)*U2(IJ)*D2P*FYS +
     &     AAN*DEN(IJP)*U2(IJP)*D2PN*FYN
      SA2N=AA*DEN(IJ)*UV(IJ)*E1P*FYS +
     &     AAN*DEN(IJP)*UV(IJP)*E1PN*FYN
      SA3N=AA*DEN(IJ)*VW(IJ)*E1P*FYS +
     &     AAN*DEN(IJP)*VW(IJP)*E1PN*FYN
      SA4N=AA*DEN(IJ)*UW(IJ)*E1P*FYS +
     &     AAN*DEN(IJP)*UW(IJP)*E1PN*FYN
      SB1N=BB*DEN(IJ)*UV(IJ)*D2P*FYS +
     &     BBN*DEN(IJP)*UV(IJP)*D2PN*FYN
      SB2N=BB*DEN(IJ)*UW(IJ)*D2P*FYS +
     &     BBN*DEN(IJP)*UW(IJP)*D2PN*FYN
      SB3N=BB*DEN(IJ)*V2(IJ)*E1P*FYS +
```

```
     &    BBN*DEN(IJP)*V2(IJP)*E1PN*FYN
      SB4N=BB*DEN(IJ)*VW(IJ)*E1P*FYS +
     &    BBN*DEN(IJP)*VW(IJP)*E1PN*FYN
C
      UWN=(UW(IJ)-TA1P-TA2P-TA3P+TA4P-
     &     TB1P-TB2P-TB3P-TB4P)*FYS +
     &   (UW(IJP)-TA1N-TA2N-TA3N+TA4N-
     &     TB1N-TB2N-TB3N-TB4N)*FYN +
     & SA1N*WPN+SA2N*WPN+SA3N*UPN-SA4N*VPN+
     & SB1N*WPN+SB2N*VPN+SB3N*WPN+SB4N*VPN
C
      UWN=UWN*RRDENN
C
C
C---CONSIDER TERMS IN VWE-EQ.----
C
      D1P=-DYEP*APVW(IJ)*RP
      D2P=DYKP*APVW(IJ)*RP
      E1P=-DXKP*APVW(IJ)*RP
      E2P=DXEP*APVW(IJ)*RP
C
      D1PE=-DYEE*APVW(IPJ)*RPE
      D2PE=DYKE*APVW(IPJ)*RPE
      E1PE=-DXKE*APVW(IPJ)*RPE
      E2PE=DXEE*APVW(IPJ)*RPE
C
      D1PN=-DYEN*APVW(IJP)*RPN
      D2PN=DYKN*APVW(IJP)*RPN
      E1PN=-DXKN*APVW(IJP)*RPN
      E2PN=DXEN*APVW(IJP)*RPN
C
      APVWE=APVW(IJ)*FXW+APVW(IPJ)*FXE
      D1E=-APVWE*DYE*RE
      D2E=APVWE*(YE-YP)*RE
      E1E=-APVWE*(XE-XP)*RE
      E2E=APVWE*DXE*RE
C
      APVWN=APVW(IJ)*FYS+APVW(IJP)*FYN
      D1N=-APVWN*(YN-YP)*RN
      D2N=APVWN*DYN*RN
      E1N=-APVWN*DXN*RN
      E2N=APVWN*(XN-XP)*RN
C
      AA=1.-C2+1.5*C2*C2P*FUNY(IJ)
      AAE=1.-C2+1.5*C2*C2P*FUNY(IPJ)
      AAN=1.-C2+1.5*C2*C2P*FUNY(IJP)
      BB=1.5*C2*C2P*FUNXY(IJ)
      BBE=1.5*C2*C2P*FUNXY(IPJ)
      BBN=1.5*C2*C2P*FUNXY(IJP)
C
      TA1P=AA*DEN(IJ)*UV(IJ)*D1P*DW1(IJ)
      TA1E=AAE*DEN(IPJ)*UV(IPJ)*D1PE*DW1(IPJ)
      TA2P=AA*DEN(IJ)*UW(IJ)*D1P*DV1(IJ)
      TA2E=AAE*DEN(IPJ)*UW(IPJ)*D1PE*DV1(IPJ)
      TA3P=AA*DEN(IJ)*V2(IJ)*E2P*DW1(IJ)
      TA3E=AAE*DEN(IPJ)*V2(IPJ)*E2PE*DW1(IPJ)
      TA4P=AA*DEN(IJ)*VW(IJ)*E2P*DV1(IJ)
      TA4E=AAE*DEN(IPJ)*VW(IPJ)*E2PE*DV1(IPJ)
      TB1P=BB*DEN(IJ)*U2(IJ)*D1P*DW1(IJ)
      TB1E=BBE*DEN(IPJ)*U2(IPJ)*D1PE*DW1(IPJ)
      TB2P=BB*DEN(IJ)*UV(IJ)*E2P*DW1(IJ)
      TB2E=BBE*DEN(IPJ)*UV(IPJ)*E2PE*DW1(IPJ)
      TB3P=BB*DEN(IJ)*VW(IJ)*E2P*DU1(IJ)
      TB3E=BBE*DEN(IPJ)*VW(IPJ)*E2PE*DU1(IPJ)
      TB4P=BB*DEN(IJ)*UW(IJ)*E2P*DV1(IJ)
      TB4E=BBE*DEN(IPJ)*UW(IPJ)*E2PE*DV1(IPJ)
C
      SA1E=AA*DEN(IJ)*UV(IJ)*D1P*FXW +
     &    AAE*DEN(IPJ)*UV(IPJ)*D1PE*FXE
      SA2E=AA*DEN(IJ)*UW(IJ)*D1P*FXW +
     &    AAE*DEN(IPJ)*UW(IPJ)*D1PE*FXE
```



```
      SA3E=AA*DEN(IJ)*V2(IJ)*E2P*FXW +
     &    AAE*DEN(IPJ)*V2(IPJ)*E2PE*FXE
      SA4E=AA*DEN(IJ)*VW(IJ)*E2P*FXW +
     &    AAE*DEN(IPJ)*VW(IPJ)*E2PE*FXE
      SB1E=BB*DEN(IJ)*U2(IJ)*D1P*FXW +
     &    BBE*DEN(IPJ)*U2(IPJ)*D1PE*FXE
      SB2E=BB*DEN(IJ)*UV(IJ)*E2P*FXW +
     &    BBE*DEN(IPJ)*UV(IPJ)*E2PE*FXE
      SB3E=BB*DEN(IJ)*VW(IJ)*E2P*FXW +
     &    BBE*DEN(IPJ)*VW(IPJ)*E2PE*FXE
      SB4E=BB*DEN(IJ)*UW(IJ)*E2P*FXW +
     &    BBE*DEN(IPJ)*UW(IPJ)*E2PE*FXE
C
      VWE=(VW(IJ)-TA1P-TA2P-TA3P-TA4P-
     &     TB1P-TB2P-TB3P+TB4P)*FXW +
     &   (VW(IPJ)-TA1E-TA2E-TA3E-TA4E-
     &     TB1E-TB2E-TB3E+TB4E)*FXE +
     & SA1E*WPE+SA2E*VPE+SA3E*WPE+SA4E*VPE+
     & SB1E*WPE+SB2E*WPE+SB3E*UPE-SB4E*VPE
C
      VWE=VWE*RRDENE
C
C---CONSIDER TERMS IN VWN-EQ.----
C
      TA1P=AA*DEN(IJ)*UV(IJ)*D2P*DW2(IJ)
      TA1N=AAN*DEN(IJP)*UV(IJP)*D2PN*DW2(IJP)
      TA2P=AA*DEN(IJ)*UW(IJ)*D2P*DV2(IJ)
      TA2N=AAN*DEN(IJP)*UW(IJP)*D2PN*DV2(IJP)
      TA3P=AA*DEN(IJ)*V2(IJ)*E1P*DW2(IJ)
      TA3N=AAN*DEN(IJP)*V2(IJP)*E1PN*DW2(IJP)
      TA4P=AA*DEN(IJ)*VW(IJ)*E1P*DV2(IJ)
      TA4N=AAN*DEN(IJP)*VW(IJP)*E1PN*DV2(IJP)
      TB1P=BB*DEN(IJ)*U2(IJ)*D2P*DW2(IJ)
      TB1N=BBN*DEN(IJP)*U2(IJP)*D2PN*DW2(IJP)
      TB2P=BB*DEN(IJ)*UV(IJ)*E1P*DW2(IJ)
      TB2N=BBN*DEN(IJP)*UV(IJP)*E1PN*DW2(IJP)
      TB3P=BB*DEN(IJ)*VW(IJ)*E1P*DU2(IJ)
      TB3N=BBN*DEN(IJP)*VW(IJP)*E1PN*DU2(IJP)
      TB4P=BB*DEN(IJ)*UW(IJ)*E1P*DV2(IJ)
      TB4N=BBN*DEN(IJP)*UW(IJP)*E1PN*DV2(IJP)
C
      SA1N=AA*DEN(IJ)*UV(IJ)*D2P*FYS +
     &    AAN*DEN(IJP)*UV(IJP)*D2PN*FYN
      SA2N=AA*DEN(IJ)*UW(IJ)*D2P*FYS +
     &    AAN*DEN(IJP)*UW(IJP)*D2PN*FYN
      SA3N=AA*DEN(IJ)*V2(IJ)*E1P*FYS +
     &    AAN*DEN(IJP)*V2(IJP)*E1PN*FYN
      SA4N=AA*DEN(IJ)*VW(IJ)*E1P*FYS +
     &    AAN*DEN(IJP)*VW(IJP)*E1PN*FYN
      SB1N=BB*DEN(IJ)*U2(IJ)*D2P*FYS +
     &    BBN*DEN(IJP)*U2(IJP)*D2PN*FYN
      SB2N=BB*DEN(IJ)*UV(IJ)*E1P*FYS +
     &    BBN*DEN(IJP)*UV(IJP)*E1PN*FYN
      SB3N=BB*DEN(IJ)*VW(IJ)*E1P*FYS +
     &    BBN*DEN(IJP)*VW(IJP)*E1PN*FYN
      SB4N=BB*DEN(IJ)*UW(IJ)*E1P*FYS +
     &    BBN*DEN(IJP)*UW(IJP)*E1PN*FYN
C
      VWN=(VW(IJ)-TA1P-TA2P-TA3P-TA4P-
     &     TB1P-TB2P-TB3P+TB4P)*FYS +
     &   (VW(IJP)-TA1N-TA2N-TA3N-TA4N-
     &     TB1N-TB2N-TB3N+TB4N)*FYN +
     & SA1N*WPN+SA2N*VPN+SA3N*WPN+SA4N*VPN+
     & SB1N*WPN+SB2N*WPN+SB3N*UPN-SB4N*VPN
C
      VWN=VWN*RRDENN
C
C---CALCULATE (U2E-U2W), (U2N-U2S) ....ETC.
C
      RE=HAF*(R(IJ)+R(IJM))
      RN=HAF*(R(IJ)+R(IMJ))
```

```
      RRW=HAF*(R(IMJ)+R(IMJ-1))
      RS=HAF*(R(IJM)+R(IMJ-1))
C
      DU2EW(IJ)=UUE-UUW(J)
      DU2NS(IJ)=UUN-UUS
      DUVNS(IJ)=UVN-UVS
      DUVEW(IJ)=UVE-UVW(J)
      DV2EW(IJ)=VVE-VVW(J)
      DV2NS(IJ)=VVN-VVS
      DUWEW(IJ)=UWE-UWW(J)
      DUWNS(IJ)=UWN-UWS
      DVWEW(IJ)=VWE-VWW(J)
      DVWNS(IJ)=VWN-VWS
C
      UUW(J)=UUE
      UVW(J)=UVE
      VVW(J)=VVE
      UWW(J)=UWE
      VWW(J)=VWE
C
320   CONTINUE
321   CONTINUE
C
      DO 330 I=2,NIM
      DO 330 J=2,NJM
      IJ=IMNJ(I)+J
      IMJ=IJ-NJ
      IJM=IJ-1
      RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))
      DPU2EW(IJ)=DU2EW(IJ)/RP
      DPU2NS(IJ)=DU2NS(IJ)/RP
      DPV2EW(IJ)=DV2EW(IJ)/RP
      DPV2NS(IJ)=DV2NS(IJ)/RP
330   CONTINUE
C
      RETURN
      END
C
C
C-----------------------------------------------------
      SUBROUTINE MODPIJ (ITBS,ITBN,JTBW,JTBE)
C-----------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'rsm.h'
      DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
C
      NI=NIM+1
      NJ=NJM+1
C
C-- SOUTHE WALL
C
      DO 1110 I=2,NIM
      IJ=IMNJ(I)+2
      IF(ITBS(I).EQ.4) THEN
      GEN(IJ)=GENTS(I)
      ENDIF
C
C---- NORTH WALL
C
      IJ=IMNJ(I)+NJM
      IF(ITBN(I).EQ.4) THEN
      GEN(IJ)=GENTN(I)
      ENDIF
C
1110  CONTINUE
C
C--- WEST WALL
C
      DO 1120 J=2,NJM
      IJ=IMNJ(2)+J
      IF(JTBW(J).EQ.4) THEN
```



```
      GEN(IJ)=GENTW(J)
      ENDIF
C
C--- EAST WALL
C
      IJ=IMNJ(NIM)+J
      IF(JTBE(J).EQ.4) THEN
      GEN(IJ)=GENTEE(J)
      ENDIF
C
1120  CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------
      SUBROUTINE MODUIUJ (PHI,IPHI,X,Y,FX,FY,ARE,VOL,R,
     &                    DEN,TE,ED,ITBS,ITBN,JTBE,JTBW,
     &                    DNS,DNN,DNE,DNW)
C-----------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'rsm.h'
      DIMENSION PHI(NXNY),DEN(NXNY),TE(NXNY),ED(NXNY)
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),ARE(NXNY),
     #          VOL(NXNY),R(NXNY)
      DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
      DIMENSION DNS(NX),DNN(NX),DNW(NY),DNE(NY)
C
C--- CONSTANTS FOR THE REYNOLDS STRESSES NEAR THE WALL
C--- TAKEN FROM (LEIN @ LESCHZINER)
C       CUU=1.098
C       CVV=0.247
C       CWW=0.655
C       CUV=-0.255
C       CVW= 0.0
C       CUW= 0.0
C-- VALUES OF CVW AND CUW ARE SET TO 0.0
C-- FOR LACK OF BETTER VALUES !!
C
C--- UPDATE  -TE-,-ED-,-U2-,-V2-,-W2-,-UV-,-VW-,-UW- B.C'S-
      GO TO (1200,1300,1400,1500,1600,1700,1800,1900) IDIR
C
C---- TURBULENT KINETIC ENERGY BOUNDARY CONDITIONS
C
1200  CONTINUE
C-----SOUTH BOUNDARY
      DO 1210 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1211,1212,1212,1213) ITBS(I)
1211  CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1212
1213  CONTINUE
      GEN(IJ)=GENTS(I)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
1212  CONTINUE
      AS(IJ)=0.0
1210  CONTINUE
C-----NORTH BOUNDARY
      DO 1220 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1221,1222,1222,1223) ITBN(I)
1221  CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1222
1223  CONTINUE
      GEN(IJ)=GENTN(I)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
```

```
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
 1222 CONTINUE
      AN(IJ)=0.0
 1220 CONTINUE
C-----WEST BOUNDARY
      DO 1230 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1231,1232,1232,1233) JTBW(J)
 1231 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*TE(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1232
 1233 CONTINUE
      GEN(IJ)=GENTW(J)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
 1232 CONTINUE
      AW(IJ)=0.0
 1230 CONTINUE
C-----EAST BOUNDARY
      DO 1240 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1241,1242,1242,1243) JTBE(J)
 1241 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*TE(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1242
 1243 CONTINUE
      GEN(IJ)=GENTEE(J)
      RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
      SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
 1242 CONTINUE
      AE(IJ)=0.0
 1240 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR TURBULENT ENERGY DISSIPATION
C
 1300 CONTINUE
C-----SOUTH BOUNDARY
      DO 1310 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1311,1312,1312,1313) ITBS(I)
 1311 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1312
 1313 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNS(I))
      BP(IJ)=GREAT
 1312 CONTINUE
      AS(IJ)=0.0
 1310 CONTINUE
C-----NORTH BOUNDARY
      DO 1320 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1321,1322,1322,1323) ITBN(I)
 1321 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1322
 1323 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNN(I))
      BP(IJ)=GREAT
 1322 CONTINUE
      AN(IJ)=0.0
 1320 CONTINUE
C-----WEST BOUNDARY
```



```
      DO 1330 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1331,1332,1332,1333) JTBW(J)
 1331 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1332
 1333 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNW(J))
      BP(IJ)=GREAT
 1332 CONTINUE
      AW(IJ)=0.0
 1330 CONTINUE
C-----EAST BOUNDARY
      DO 1340 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1341,1342,1342,1343) JTBE(J)
 1341 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1342
 1343 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNE(J))
      BP(IJ)=GREAT
 1342 CONTINUE
      AE(IJ)=0.0
 1340 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR U2-REYNOLDS STRESS
C
 1400 CONTINUE
C-----SOUTH BOUNDARY
      DO 1410 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1411,1412,1412,1413) ITBS(I)
 1411 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*U2(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1412
 1413 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UUREAL=UUWAL*FT1S(I)**2+VVWAL*FN1S(I)**2+
     &        2.*UVWAL*FT1S(I)*FN1S(I)
      SU(IJ)=GREAT*UUREAL
      BP(IJ)=GREAT
 1412 CONTINUE
      AS(IJ)=0.
 1410 CONTINUE
C-----NORTH BOUNDARY
      DO 1420 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1421,1422,1422,1423) ITBN(I)
 1421 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*U2(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1422
 1423 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UUREAL=UUWAL*FT1N(I)**2+VVWAL*FN1N(I)**2+
     &        2.*UVWAL*FT1N(I)*FN1N(I)
      SU(IJ)=GREAT*UUREAL
```

```
      BP(IJ)=GREAT
 1422 CONTINUE
      AN(IJ)=0.
 1420 CONTINUE
C-----WEST BOUNDARY
      DO 1430 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1431,1432,1432,1433) JTBW(J)
 1431 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*U2(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1432
 1433 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UUREAL=UUWAL*FT1W(J)**2+VVWAL*FN1W(J)**2+
     &        2.*UVWAL*FT1W(J)*FN1W(J)
      SU(IJ)=GREAT*UUREAL
      BP(IJ)=GREAT
 1432 CONTINUE
      AW(IJ)=0.
 1430 CONTINUE
C-----EAST BOUNDARY
      DO 1440 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1441,1442,1442,1443) JTBE(J)
 1441 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*U2(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1442
 1443 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UUREAL=UUWAL*FT1E(J)**2+VVWAL*FN1E(J)**2+
     &        2.*UVWAL*FT1E(J)*FN1E(J)
      SU(IJ)=GREAT*UUREAL
      BP(IJ)=GREAT
 1442 CONTINUE
      AE(IJ)=0.
 1440 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR V2-REYNOLDS STRESS
C
 1500 CONTINUE
C-----SOUTH BOUNDARY
      DO 1510 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1511,1512,1512,1513) ITBS(I)
 1511 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*V2(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1512
 1513 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      VVREAL=UUWAL*FT2S(I)**2+VVWAL*FN2S(I)**2+
     &        2.*UVWAL*FT2S(I)*FN2S(I)
      SU(IJ)=GREAT*VVREAL
      BP(IJ)=GREAT
 1512 CONTINUE
      AS(IJ)=0.
 1510 CONTINUE
C-----NORTH BOUNDARY
```



```
      DO 1520 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1521,1522,1522,1523) ITBN(I)
 1521 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*V2(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1522
 1523 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      VVREAL=UUWAL*FT2N(I)**2+VVWAL*FN2N(I)**2+
     &        2.*UVWAL*FT2N(I)*FN2N(I)
      SU(IJ)=GREAT*VVREAL
      BP(IJ)=GREAT
 1522 CONTINUE
      AN(IJ)=0.
 1520 CONTINUE
C-----WEST BOUNDARY
      DO 1530 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1531,1532,1532,1533) JTBW(J)
 1531 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*V2(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1532
 1533 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      VVREAL=UUWAL*FT2W(J)**2+VVWAL*FN2W(J)**2+
     &        2.*UVWAL*FT2W(J)*FN2W(J)
      SU(IJ)=GREAT*VVREAL
      BP(IJ)=GREAT
 1532 CONTINUE
      AW(IJ)=0.
 1530 CONTINUE
C-----EAST BOUNDARY
      DO 1540 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1541,1542,1542,1543) JTBE(J)
 1541 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*V2(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1542
 1543 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      VVREAL=UUWAL*FT2E(J)**2+VVWAL*FN2E(J)**2+
     &        2.*UVWAL*FT2E(J)*FN2E(J)
      SU(IJ)=GREAT*VVREAL
      BP(IJ)=GREAT
 1542 CONTINUE
      AE(IJ)=0.
 1540 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR W2-REYNOLDS STRESS
C
 1600 CONTINUE
C-----SOUTH BOUNDARY
      DO 1610 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1611,1612,1612,1613) ITBS(I)
 1611 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*W2(IJ-1)
```

```
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1612
 1613 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CWW*TE(IJ)
      BP(IJ)=GREAT
 1612 CONTINUE
      AS(IJ)=0.
 1610 CONTINUE
C-----NORTH BOUNDARY
      DO 1620 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1621,1622,1622,1623) ITBN(I)
 1621 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*W2(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1622
 1623 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CWW*TE(IJ)
      BP(IJ)=GREAT
 1622 CONTINUE
      AN(IJ)=0.
 1620 CONTINUE
C-----WEST BOUNDARY
      DO 1630 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1631,1632,1632,1633) JTBW(J)
 1631 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*W2(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1632
 1633 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CWW*TE(IJ)
      BP(IJ)=GREAT
 1632 CONTINUE
      AW(IJ)=0.
 1630 CONTINUE
C-----EAST BOUNDARY
      DO 1640 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1641,1642,1642,1643) JTBE(J)
 1641 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*W2(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1642
 1643 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      SU(IJ)=GREAT*CWW*TE(IJ)
      BP(IJ)=GREAT
 1642 CONTINUE
      AE(IJ)=0.
 1640 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR UV-REYNOLDS STRESS
C
 1700 CONTINUE
C-----SOUTH BOUNDARY
      DO 1710 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1711,1712,1712,1713) ITBS(I)
 1711 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*UV(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1712
 1713 CONTINUE
      DXB=XX(IJ-1)-XX(IJ-NJ-1)
      DYB=YY(IJ-1)-YY(IJ-NJ-1)
```



```
      ARW=SQRT(DXB**2+DYB**2)+SMALL
      DXB=DXB/ARW
      DYB=DYB/ARW
      VP2=U(IJ)*DXB+V(IJ)*DYB
      GRADVP2=ABS(VP2)-UWALL
      SIGN=1.0
      IF(GRADVP2.LT.0.) SIGN=-1.0
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UVREAL=UUWAL*FT1S(I)*FT2S(I)+VVWAL*FN1S(I)*FN2S(I)+
     &       UVWAL*(FT1S(I)*FN2S(I)+FT2S(I)*FN1S(I))
      SU(IJ)=GREAT*SIGN*UVREAL
      BP(IJ)=GREAT
 1712 CONTINUE
      AS(IJ)=0.
 1710 CONTINUE
C-----NORTH BOUNDARY
      DO 1720 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1721,1722,1722,1723) ITBN(I)
 1721 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*UV(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1722
 1723 CONTINUE
      DXB=XX(IJ)-XX(IJ-NJ)
      DYB=YY(IJ)-YY(IJ-NJ)
      ARW=SQRT(DXB**2+DYB**2)+SMALL
      DXB=DXB/ARW
      DYB=DYB/ARW
      VP2=U(IJ)*DXB+V(IJ)*DYB
      GRADVP2=UWALL-ABS(VP2)
      SIGN=1.0
      IF(GRADVP2.LT.0.) SIGN=-1.0
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UVREAL=UUWAL*FT1N(I)*FT2N(I)+VVWAL*FN1N(I)*FN2N(I)+
     &       UVWAL*(FT1N(I)*FN2N(I)+FT2N(I)*FN1N(I))
      SU(IJ)=GREAT*SIGN*UVREAL
      BP(IJ)=GREAT
 1722 CONTINUE
      AN(IJ)=0.
 1720 CONTINUE
C-----WEST BOUNDARY
      DO 1730 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1731,1732,1732,1733) JTBW(J)
 1731 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*UV(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1732
 1733 CONTINUE
      DXB=XX(IJ-NJ)-XX(IJ-NJ-1)
      DYB=YY(IJ-NJ)-YY(IJ-NJ-1)
      ARW=SQRT(DXB**2+DYB**2)+SMALL
      DXB=DXB/ARW
      DYB=DYB/ARW
      VP2=U(IJ)*DXB+V(IJ)*DYB
      GRADVP2=ABS(VP2)-VWALL
      SIGN=1.0
      IF(GRADVP2.LT.0.) SIGN=-1.0
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UVREAL=UUWAL*FT1W(J)*FT2W(J)+VVWAL*FN1W(J)*FN2W(J)+
     &       UVWAL*(FT1W(J)*FN2W(J)+FT2W(J)*FN1W(J))
```

```
      SU(IJ)=GREAT*SIGN*UVREAL
      BP(IJ)=GREAT
 1732 CONTINUE
      AW(IJ)=0.
 1730 CONTINUE
C-----EAST BOUNDARY
      DO 1740 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1741,1742,1742,1743) JTBE(J)
 1741 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*UV(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1742
 1743 CONTINUE
      DXB=XX(IJ)-XX(IJ-1)
      DYB=YY(IJ)-YY(IJ-1)
      ARW=SQRT(DXB**2+DYB**2)+SMALL
      DXB=DXB/ARW
      DYB=DYB/ARW
      VP2=U(IJ)*DXB+V(IJ)*DYB
      GRADVP2=VWALL-ABS(VP2)
      SIGN=1.0
      IF(GRADVP2.LT.0.) SIGN=-1.0
      TE(IJ)=ABS(TE(IJ))
      UUWAL=CUU*TE(IJ)
      VVWAL=CVV*TE(IJ)
      UVWAL=CUV*TE(IJ)
      UVREAL=UUWAL*FT1E(J)*FT2E(J)+VVWAL*FN1E(J)*FN2E(J)+
     &         UVWAL*(FT1E(J)*FN2E(J)+FT2E(J)*FN1E(J))
      SU(IJ)=GREAT*SIGN*UVREAL
      BP(IJ)=GREAT
 1742 CONTINUE
      AE(IJ)=0.
 1740 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR VW-REYNOLDS STRESS
C
 1800 CONTINUE
C-----SOUTH BOUNDARY
      DO 1810 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1811,1812,1812,1813) ITBS(I)
 1811 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*VW(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1812
 1813 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      VWREAL=UWWAL*FT2S(I)+VWWAL*FN2S(I)
      SU(IJ)=GREAT*VWREAL
      BP(IJ)=GREAT
 1812 CONTINUE
      AS(IJ)=0.
 1810 CONTINUE
C-----NORTH BOUNDARY
      DO 1820 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1821,1822,1822,1823) ITBN(I)
 1821 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*VW(IJ+1)
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1822
 1823 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      VWREAL=UWWAL*FT2N(I)+VWWAL*FN2N(I)
```



```
      SU(IJ)=GREAT*VWREAL
      BP(IJ)=GREAT
 1822 CONTINUE
      AN(IJ)=0.
 1820 CONTINUE
C-----WEST BOUNDARY
      DO 1830 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1831,1832,1832,1833) JTBW(J)
 1831 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*VW(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1832
 1833 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      VWREAL=UWWAL*FT2W(J)+VWWAL*FN2W(J)
      SU(IJ)=GREAT*VWREAL
      BP(IJ)=GREAT
 1832 CONTINUE
      AW(IJ)=0.
 1830 CONTINUE
C-----EAST BOUNDARY
      DO 1840 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1841,1842,1842,1843) JTBE(J)
 1841 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*VW(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1842
 1843 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      VWREAL=UWWAL*FT2E(J)+VWWAL*FN2E(J)
      SU(IJ)=GREAT*VWREAL
      BP(IJ)=GREAT
 1842 CONTINUE
      AE(IJ)=0.
 1840 CONTINUE
C
      RETURN
C
C-----BOUNDARY CONDITIONS FOR UW-REYNOLDS STRESS
C
 1900 CONTINUE
C-----SOUTH BOUNDARY
      DO 1910 I=2,NIM
      IJ=IMNJ(I)+2
      GO TO (1911,1912,1912,1913) ITBS(I)
 1911 CONTINUE
      SU(IJ)=SU(IJ)+AS(IJ)*UW(IJ-1)
      BP(IJ)=BP(IJ)+AS(IJ)
      GO TO 1912
 1913 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      UWREAL=UWWAL*FT1S(I)+VWWAL*FN1S(I)
      SU(IJ)=GREAT*UWREAL
      BP(IJ)=GREAT
 1912 CONTINUE
      AS(IJ)=0.
 1910 CONTINUE
C-----NORTH BOUNDARY
      DO 1920 I=2,NIM
      IJ=IMNJ(I)+NJM
      GO TO (1921,1922,1922,1923) ITBN(I)
 1921 CONTINUE
      SU(IJ)=SU(IJ)+AN(IJ)*UW(IJ+1)
```

```
      BP(IJ)=BP(IJ)+AN(IJ)
      GO TO 1922
 1923 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      UWREAL=UWWAL*FT1N(I)+VWWAL*FN1N(I)
      SU(IJ)=GREAT*UWREAL
      BP(IJ)=GREAT
 1922 CONTINUE
      AN(IJ)=0.
 1920 CONTINUE
C-----WEST BOUNDARY
      DO 1930 J=2,NJM
      IJ=IMNJ(2)+J
      GO TO (1931,1932,1932,1933) JTBW(J)
 1931 CONTINUE
      SU(IJ)=SU(IJ)+AW(IJ)*UW(IJ-NJ)
      BP(IJ)=BP(IJ)+AW(IJ)
      GO TO 1932
 1933 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      UWREAL=UWWAL*FT1W(J)+VWWAL*FN1W(J)
      SU(IJ)=GREAT*UWREAL
      BP(IJ)=GREAT
 1932 CONTINUE
      AW(IJ)=0.
 1930 CONTINUE
C-----EAST BOUNDARY
      DO 1940 J=2,NJM
      IJ=IMNJ(NIM)+J
      GO TO (1941,1942,1942,1943) JTBE(J)
 1941 CONTINUE
      SU(IJ)=SU(IJ)+AE(IJ)*UW(IJ+NJ)
      BP(IJ)=BP(IJ)+AE(IJ)
      GO TO 1942
 1943 CONTINUE
      TE(IJ)=ABS(TE(IJ))
      UWWAL=CUW*TE(IJ)
      VWWAL=CVW*TE(IJ)
      UWREAL=UWWAL*FT1E(J)+VWWAL*FN1E(J)
      SU(IJ)=GREAT*UWREAL
      BP(IJ)=GREAT
 1942 CONTINUE
      AE(IJ)=0.
 1940 CONTINUE
C
      RETURN
      END
C
C*****************************************************
      SUBROUTINE ASOLSIP(PHI,IPHI,RESOR)
C*****************************************************
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
      INCLUDE 'gridparam.h'
      INCLUDE 'rsm.h'
C
      DIMENSION PHI(NXNY), RES(NXNY)
      DIMENSION BS(NXNY), BN(NXNY), BE(NXNY), BW(NXNY)
      DIMENSION FP(NXNY), RESOR(2)
C
      NJ = NJM + 1
      NI = NIM + 1
      NINJ = NI*NJ
      DO 5 IJ=1,NINJ
         BN(IJ) = 0.
         BE(IJ) = 0.
```



```
         RES(IJ) = 1.
         FP(IJ)=PHI(IJ)
    5 CONTINUE
C
C

      DO 10 I=2,NIM
      DO 10 J=2,NJM
         IJ=IMNJ(I)+J
         API=1.0/AP(IJ)
         AP(IJ)=1.0
         AE(IJ)=AE(IJ)*API
         AW(IJ)=AW(IJ)*API
         AN(IJ)=AN(IJ)*API
         AS(IJ)=AS(IJ)*API
         SU(IJ)=SU(IJ)*API
   10 CONTINUE
C
      DO 20 I=2,NIM
      DO 20 J=2,NJM
         IJ=IMNJ(I)+J
         IJM=IJ-1
         IMJ=IJ-NJ
         BW(IJ)=-AW(IJ)/(1.+ALFAKE*BN(IJ-NJ))
         BS(IJ)=-AS(IJ)/(1.+ALFAKE*BE(IJM))
         POM1=ALFAKE*BW(IJ)*BN(IMJ)
         POM2=ALFAKE*BS(IJ)*BE(IJM)
         BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IMJ)-BS(IJ)*BN(IJM)
         BN(IJ)=(-AN(IJ)-POM1)/(BP(IJ)+SMALL)
         BE(IJ)=(-AE(IJ)-POM2)/(BP(IJ)+SMALL)
   20 CONTINUE
C
      DO 100 L=1,NSWPKE(IPHI)
       RESORP=0.
       DO 30 I=2,NIM
       DO 30 J=2,NJM
       IJ=IMNJ(I)+J
       RES(IJ)=AN(IJ)*PHI(IJ+1)+AS(IJ)*PHI(IJ-1)+AE(IJ)*PHI(IJ+NJ)+
     &     AW(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)
          RESORP=RESORP+ABS(RES(IJ))
          RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/
     &               (BP(IJ)+SMALL)
   30    CONTINUE
C
         IF(L.EQ.1) RESORKE(IPHI)=RESORP
         RSM=SORKE(IPHI)*RESORKE(IPHI)
         DO 40 I=2,NIM
            II=NIM+2-I
            DO 40 J=2,NJM
               JJ=NJM+2-J
               IJ=IMNJ(II)+JJ
               RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)
               PHI(IJ)=PHI(IJ)+RES(IJ)
   40    CONTINUE
co       IF(RESORP.LE.RSM) RETURN
         IF(RESORP.LE.RSM) GOTO 200
  100 CONTINUE
C
      IF(RESORP.GE.RSM.AND.L.GE.NSWPKE(IPHI)) WRITE(*,2)
    2 FORMAT(10X,' SOLSIP DID NOT CONVERGE  ')
C
 200  CONTINUE
      AUX1=0.
      AUX2=0.
      DO 50 I=2,NIM
      DO 50 J=2,NJM
      IJ=IMNJ(I)+J
      AUX1=AUX1+ABS(PHI(IJ)-FP(IJ))
      AUX2=AUX2+ABS(FP(IJ))
 50   CONTINUE
c     IF(AUX2.LT.1.E-30) AUX2=1.E-30
```

```
      RESOR(IPHI)=AUX1/AUX2
      RETURN
      END
C
C.....A GAUSS ELIMINATION SOLVER
      SUBROUTINE SOLV(A,BB,N)
      DIMENSION A(N,N),B(N),C(N),BB(N),X(N),MME(N)
      EP =1.E-19
      DO 10 J=1,N
10    MME(J)=J
      DO 20 I=1,N
      Y=0.
      DO 30 J=I,N
      IF(ABS(A(I,J)).LT.ABS(Y)) GOTO 30
      K=J
      Y=A(I,J)
30    CONTINUE
C
      IF(ABS(Y).LT.EP) THEN
      WRITE(*,*)
      WRITE(9,*)
      DO 35 IA=1,N
      WRITE(*,1000) (A(IA,JA),JA=1,N)
35    CONTINUE
      PRINT*,'THERE IS NO CONVERSE MATRIX'
      STOP 2222
      ENDIF
C
      Y=1./Y
      DO 40 J=1,N
      C(J)=A(J,K)
      A(J,K)=A(J,I)
      A(J,I)=-C(J)*Y
      B(J)=A(I,J)*Y
40    A(I,J)=A(I,J)*Y
      A(I,I)=Y
      J=MME(I)
      MME(I)=MME(K)
      MME(K)=J
      DO 11 K=1,N
      IF(K.EQ.I) GOTO 11
      DO 12 J=1,N
      IF(J.EQ.I) GOTO 12
      A(K,J)=A(K,J)-B(J)*C(K)
12    CONTINUE
11    CONTINUE
20    CONTINUE
      DO 33 I=1,N
      DO 44 K=1,N
      IF(MME(K).EQ.I) GOTO 55
44    CONTINUE
55    IF(K.EQ.I) GOTO 33
      DO 66 J=1,N
      W=A(I,J)
      A(I,J)=A(K,J)
66    A(K,J)=W
      IW=MME(I)
      MME(I)=MME(K)
      MME(K)=IW
33    CONTINUE
1000  FORMAT(4X,1P5E13.4)
      DO 50 I=1,N
      X(I)=0.
      DO 50 J=1,N
50    X(I)=X(I)+A(I,J)*BB(J)
      DO 60 I=1,N
60    BB(I)=X(I)
      RETURN
      END
C-----------------------------------------------------
      SUBROUTINE AMODIFY(SUASM,SVASM,SWASM,
```



```
     &  X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,DEN,
     &  VISCOS,TE,DNS,DNN,DNW,DNE,ITBS,ITBN,JTBW,JTBE,
     &  U,V,W,ITER)
C-----------------------------------------------------
      INCLUDE 'gridparam.h'
      INCLUDE 'rsm.h'
C
      DIMENSION UUW(NY),VVW(NY),UVW(NY),UWW(NY),VWW(NY)
      DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),R(NXNY)
      DIMENSION DNS(NX),DNN(NX),DNW(NY),DNE(NY)
      DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY),TE(NXNY)
      DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)
      IF(ITER.EQ.1) THEN
      REWIND 41
      READ(41,*)
      READ(41,*)
      READ(41,*) CD1,CD2,CMU,ELOG,CAPPA
      READ(41,*)
      READ(41,*) LRE,LAY2
      END IF
C
      NI = NIM + 1
      NJ = NJM + 1
      CMU25=SQRT(SQRT(CMU))
C
C-----SOUTH BOUNDARY
      DO 600 I=2,NIM
         IJ=IMNJ(I)+2
         IF(ITBS(I).EQ.4) THEN
         DXB=X(IJ-1)-X(IJ-NJ-1)
         DYB=Y(IJ-1)-Y(IJ-NJ-1)
         ARW=SQRT(DXB**2+DYB**2)
         DXB=DXB/ARW
         DYB=DYB/ARW
         CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
         YPLS=DNS(I)*CONST/VISCOS
         IF (YPLS.LE.11.63.OR.LAY2) THEN
         TCOEF=VISCOS/DNS(I)
         ELSE
         UPLUS=LOG(ELOG*YPLS)/CAPPA
         TCOEF=CONST/UPLUS
         ENDIF
         VPINT=U(IJ)*DXB+V(IJ)*DYB
         VPINT=VPINT+W(IJ)
         VPINT=ABS(VPINT-SQRT(U(IJ-1)*U(IJ-1)+
     1         V(IJ-1)*V(IJ-1)+W(IJ-1)*W(IJ-1)))
         GENTS(I)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNS(I))
C
         ENDIF
C-----NORTH BOUNDARY
         IJ=IMNJ(I)+NJM
         IF (ITBN(I).EQ.4) THEN
         DXB=X(IJ)-X(IJ-NJ)
         DYB=Y(IJ)-Y(IJ-NJ)
         ARW=SQRT(DXB**2+DYB**2)
         DXB=DXB/ARW
         DYB=DYB/ARW
         CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
         YPLS=DNN(I)*CONST/VISCOS
         IF (YPLS.LE.11.63.OR.LAY2) THEN
         TCOEF=VISCOS/DNN(I)
         ELSE
         UPLUS=LOG(ELOG*YPLS)/CAPPA
         TCOEF=CONST/UPLUS
         ENDIF
         VPINT=U(IJ)*DXB+V(IJ)*DYB
         VPINT=VPINT+W(IJ)
         VPINT=ABS(VPINT-SQRT(U(IJ+1)*U(IJ+1)+
     1         V(IJ+1)*V(IJ+1)+W(IJ+1)*W(IJ+1)))
         GENTN(I)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNN(I))
C
```

```
            ENDIF
  600 CONTINUE
C-----WEST BOUNDARY
      DO 620 J=2,NJM
      IJ=IMNJ(2)+J
      IF (JTBW(J).EQ.4) THEN
      DXB=X(IJ-NJ)-X(IJ-NJ-1)
      DYB=Y(IJ-NJ)-Y(IJ-NJ-1)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
      YPLS=DNW(J)*CONST/VISCOS
      IF (YPLS.LE.11.63.OR.LAY2) THEN
      TCOEF=VISCOS/DNW(J)
      ELSE
      UPLUS=LOG(ELOG*YPLS)/CAPPA
      TCOEF=CONST/UPLUS
      ENDIF
      VPINT=U(IJ)*DXB+V(IJ)*DYB
      VPINT=VPINT+W(IJ)
      VPINT=ABS(VPINT-SQRT(U(IJ-NJ)*U(IJ-NJ)+
     1           V(IJ-NJ)*V(IJ-NJ)+W(IJ-NJ)*W(IJ-NJ)))
      GENTW(J)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNW(J))
      ENDIF
C-----EAST BOUNDARY
      IJ=IMNJ(NIM)+J
      IF (JTBE(J).EQ.4) THEN
      UP=U(IJ)
      VP=V(IJ)
      WP=W(IJ)
      UWALL=U(IJ+NJ)
      VWALL=V(IJ+NJ)
      WWALL=W(IJ+NJ)
      TEPR=SQRT(TE(IJ))
      DELN=DNE(J)
      RB=HAF*(R(IJ)+R(IJ-1))
      DENS=DEN(IJ)
      DXB=X(IJ)-X(IJ-1)
      DYB=Y(IJ)-Y(IJ-1)
      ARW=SQRT(DXB**2+DYB**2)
      DXB=DXB/ARW
      DYB=DYB/ARW
      CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
      YPLS=DNE(J)*CONST/VISCOS
      IF (YPLS.LE.11.63.OR.LAY2) THEN
      TCOEF=VISCOS/DNE(J)
      ELSE
      UPLUS=LOG(ELOG*YPLS)/CAPPA
      TCOEF=CONST/UPLUS
      ENDIF
      VPINT=U(IJ)*DXB+V(IJ)*DYB
      VPINT=VPINT+W(IJ)
      VPINT=ABS(VPINT-SQRT(U(IJ+NJ)*U(IJ+NJ)+
     1           V(IJ+NJ)*V(IJ+NJ)+W(IJ+NJ)*W(IJ+NJ)))
      GENTEE(J)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNE(J))
      ENDIF
  620 CONTINUE
C
      RETURN
      END
C
C--------------------------------------------------------
C  gridparam.h
C
      PARAMETER (HAF=0.5,QTR=0.25)
      PARAMETER (SMALL=1.E-30,GREAT=1.E30)
      PARAMETER (NX=100)
      PARAMETER (NY=50)
      PARAMETER (NXNY=NX*NY)
C
```



```
C--------------------------------------------------------
C rsm.h
      PARAMETER (HAF=0.5, QTR=0.25)
      PARAMETER (SMALL=1.0E-30, GREAT=1.0E+30)
C
      COMMON /A1/ AS(NXNY), AN(NXNY), AE(NXNY), AW(NXNY),
     1            AP(NXNY), BP(NXNY), SU(NXNY),APU(NXNY),APV(NXNY)
C
      COMMON /A2/ SORKE(2), NSWPKE(2), URFKE(2),PRTKE(2)
C
      COMMON /A3/ GKE,ALFAKE,RESORKE(2),URFVIS
      COMMON /CONSTT/ CMU,CMU75,ELOG,CAPPA,
     &                C1,C2,C1P,C2P,CK,CE,CD1,CD2
      COMMON /A3/ GENTS(NX),GENTN(NX),GENTW(NY),GENTEE(NY),P11(NXNY),
     1            P22(NXNY),P33(NXNY),P12(NXNY),P13(NXNY),P23(NXNY)
      COMMON /A4/ FUNX(NXNY),FUNY(NXNY),FUNXY(NXNY),
     1            GEN(NXNY),DUDX(NXNY),DUDY(NXNY),DVDX(NXNY),
     2            DVDY(NXNY),DWDX(NXNY),DWDY(NXNY)
      COMMON /STRESS/ U2(NXNY),V2(NXNY),W2(NXNY),
     1                UV(NXNY),VW(NXNY),UW(NXNY)
C
      LOGICAL AKSI,RESTART
```

CHAPTER 6

3D k -ε Turbulence Model

Table of Contents

6.1 Introduction

In this section a description of the standard k-ε turbulence model that is coded as a self contained computer program to compute turbulent flow quantities in three-dimensional, body-fitted geometry is given. Module structure and variables used are given in the Appendix. The module was successfully tested as a self-contained unit using the REACT code[1].

6.2 Theory and Model Equations

The k-ε turbulence model used is based on the standard two equation k-ε model of Launder and Splading [2]. For a steady, incompressible flow the transport equations for the turbulent kinetic energy k and energy dissipation ε can be written in generalized Cartesian coordinates as;

$$\frac{\partial \rho U_j k}{\partial x_j} = \frac{1}{r}\frac{\partial}{\partial x_j}(\mu + \frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial xj}) + G - \rho\,\varepsilon \tag{1}$$

$$\frac{\partial \rho U_j \varepsilon}{\partial x_j} = \frac{1}{r}\frac{\partial}{\partial x_j}(\mu + \frac{\mu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial xj}) + G - \rho\,\varepsilon \tag{2}$$

where G denotes the rate of production of turbulent kinetic energy and is expressed as;

$$G = \mu_t \frac{\partial U_i}{\partial x_j}(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i})$$

The empirical constants, σ_k, σ_ε, C_1 and C_2 have values 1.0, 1.0, 1.44 and 1.92 respectively.

The above equations are valid only in the fully turbulent region away from the wall. Therefore the wall function method (similar to that described in Chapter 2 for the 2D k-ε module) is used to model the damping effects of the thin sublayer region close to the wall.

6.3 Module Evaluation

The 3D k-ε turbulent module was evaluated by interfacing the module with the REACT code as the CFD solver and producing the same results that were generated previously with the full REACT code for a centrifugal impeller calculations (Chen et. al [3]). Figure 1 shows the grid topology of the impeller studied with the shroud removed and Figure 2, shows the reduced pressure plot . In general Chen et al's calculations showed good comparisons with experimental data obtained from laser velocimetry in a water test rig.

REFERENCES

[1] Darian, A. and Chan, D "A User's Manual for the Rocketdyne Elliptic Analysis Code for Turbomachinery" CFD Technology Center, Rocketdyne Division/Rockwell International, 1992.

[2] Launder, B. E. and Spalding, D. B. "The Numerical Computation of Turbulent Flows" Computer Methods in Applied Mechanics and Engineering, vol. 3, pp. 269-289, 1974.

[3] Chen, W, Eastland, A., Brozowski, A. and Darian A. "A Comparison of Numerical and Experimental Results of a Centrifugal Impeller Flow" Fifth Int. Symp. on Transport Phenomena and Dynamics of Rotating Machinery, Kaanapali, Hawaii, 1992.

[4] Stone, H. "Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations" SIAM J. Num. Anal., vol. 5, pp. 530 - , 1968.

Figure 1. Impeller grid topology

Figure 2. Reduced pressures

APPENDIX E

3D k-ε Turbulence Module Deck

This module consists of two separate programs KEMOD3 and MODIFY, which have to be linked to the main flow solver. A description of each file will be given next.

Program KEMOD3

This is basically the solver for the k and ε - transport equations. It reads through its argument list different variables from the calling flow solver. These variables are described below.

List of Argument Variable Names

NIM	Number of cell nodes in the I- or ξ-coordinate lines. (input from the flow solver)
NJM	Number of cell nodes in the J- or η-coordinate lines. (input from the flow solver)
NJM	Number of cell nodes in the k- or ζ-coordinate lines. (input from the flow solver)
X	Grid node locations in the x or ξ-direction, dimensioned to X(JXYZ) (JXYZ=NX*NY*NZ) (input from flow solver)
Y	Grid node locations in the y or η-direction, dimensioned to Y(JXYZ) (input from flow solver)
Y	Grid node locations in the z or ζ-direction, dimensioned to Y(JXYZ) (input from flow solver)
U	x-direction velocity (u), dimensioned as U(JXYZ) (input from flow solver)
V	y-direction velocity (v), also dimensional as V(JXYZ) (input from flow solver)
W	z-direction velocity (w), dimensional W(JXYZ) (input from flow solver)
TE	Turbulence kinetic energy k, dimensioned TE(JXYZ) (calculated in the module and returned to the flow solver)

ED	Turbulent energy dissipation rate ε, dimensioned ED(JXYZ) (calculated in the module and returned to the flow solver)
URFK	Under-relaxation factor for k -equation (input from flow solver)
URFE	Under-relaxation factor for ε-equation (input from flow solver)
PRTK	Prandtl/Schmidt number for turbulent energy-equation, assumed known (input from flow solver)
PRTE	Prandtl/Schmidt number for turbulent energy dissipation equation, assumed known (input from flow solver)
G	= 1.0 if second order upwinding is desired = 0.0 if first order upwinding is used (input from flow solver. Usually calculation of k and ε are not very sensitive to the order of upwinding used)
F1	Mass flux variable at cell faces in x- or ξ-direction, dimensioned F1(JXYZ) (input from flow solver)
F2	Mass flux variable at cell faces in y or η-direction, dimensioned F2(JXYZ) (input from flow solver)
F3	Mass flux variable at cell faces in z or ζ-direction, dimensioned F3(JXYZ) (input from flow solver)
ITER	Iteration number (input from flow solver)
VISCOS	Dynamic viscosity (input from flow solver)
VIS	Eddy viscosity, dimensioned VIS(JXYZ) (calculated in the module and returned to the main solver)
C1	Turbulence model constant, C_1 (input from flow solver)
C2	Turbulence model constant, C_2 (input from flow solver)
CMU	Turbulence model constant, C_μ (input from flow solver)
BCFE	Boundary condition flag along east boundary (or y-z plane). It must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set IBCE to (NY*NZ)*2, and similarly for other boundaries, dimensioned BCFE(JYZ=NY*NZ) (input from flow solver)
BCFW	Boundary condition flag along west boundary, dimensioned BCFW(JYZ) (input from flow solver)
BCFS	Boundary condition flag along the south boundary, dimensioned BCFS(JXZ=NX*NZ) (input from flow solver)

BCFN	Boundary condition flag along north boundary, dimensioned BCFN(JXZ) (input from flow solver)
BCFB	Boundary condition flag along bottom boundary (or x-y plane), dimensioned BCFB(JXY=NX*NY) (input from flow solver)
BCFT	Boundary condition flag along top boundary (or x-y plane), dimensioned BCFT(JXY=NX*NY) (input from flow solver)

The module is interfaced with the main flow solver by a call to KEMOD3 with its arguments. For iterative flow solvers the module is called within the iteration sequence after the solution of the momentum equations where the mean velocities are passed to the module. There are different flow solvers utilizing different schemes from staggered to nonstaggered grid arrangement and for nonorthogonal coordinate system there are at least three alternatives to the choice of the velocity components;

i. Cartesian velocity components
ii. Contravariant velocity components
iii. Covariant velocity components

The Cartesian velocity components are the most widely used and have the advantage of simple formulation of the governing equations. Whatever the arrangement used, mass fluxes at cell faces are required and passed to the module as F1, F2 and F3 in all directions. The location of other variables such as k and ε are at the cell center or cell nodes.

The module starts by reassigning variable names passed to it from flow solver to names that are shared with the different subroutines of the module in a common statement file "kemod.h". Then a check is made if it is the first iteration in which case the grid file "GRIDG" is called -after passing the grid node locations X, Y and Z- in order to calculate grid related quantities which will be explained later. The need to call GRIDG can be waived if all the grid data are passed to the module. That is all the information about the grid such as interpolation factors FX, FY and FZ, cell volumes (VOL) and normal distances of first grid point from grid boundaries (DNS from south boundary, DNN - from north boundary, DNW - from west boundary, DNE - from east boundary, DNB - from bottom boundary and DNT - from top boundary).

After this a call to subroutine CALCE is made to calculate the turbulent kinetic energy k (with the identifier IPHI=1). The energy dissipation equation is solved next by a call to subroutine CALCE again with the identifier IPHI=2. The turbulent viscosity is updated next by calling subroutine MODVIS. A brief description of each subroutine is given next.

Subroutine GRIDG

Before calling this subroutine, the coordinates of all grid nodes, defined in reference to a fixed Cartesian coordinate frame are read. Figure 3 shows the position of cell and grid nodes. The west-to-east, south-to-north and bottom-to-top directions correspond to the ascending indexing order of i, j and k, respectively, forming a right-handed coordinate system.

This subroutine is called only once to calculate coordinates of grid nodes (intersection of grid lines) and geometrical properties of the grid (cell volumes, interpolation factors, normal distances of near-boundary cell nodes from boundary). All variables including grid node coordinates are converted to one-dimensional arrays. The position of any node in one-dimensional array is therefore defined as;

IJK = (I-1)*NJ + (K-1)*NI*NJ + J

where NI, NJ and NK are the maximum number of grid nodes in the i, j and k directions respectively.

The actual number of grid nodes is one row and one column less than for all cell nodes. For I = NI, J = NJ and K = NK fictitious grid nodes are introduced which have the same coordinates as actual nodes on NI-1 in I-direction, NJ-1 in the J-direction and NK-1 in K-direction.

The subroutine then calculates interpolation factors which are associated with cell nodes and are used in the main program to calculate values of dependent variables at locations other than cell nodes (cell centers). Cell volumes are calculated next followed by calculations of normal distances of near-boundary nodes from all the six outer boundaries.

Subroutine CALCE (PHI, IPHI)

This subroutine solves the linearized and discretized transport equations for the turbulent energy k and the energy dissipation rate ε. The two dummy parameters in the calling statement, PHI and IPHI, represent arrays containing dependent variables for which the equation is to be solved. The subroutine sets up the convective and diffusive coefficients over the entire field, then it calculates the source terms for either k or ε transport equations. A call is made to MODKE or MODED in order to modify the sources for k and ε equations respectively.

The discretized equations have the form

$$A_p \Phi_p = \sum_{i=EWNSTB} A_i \Phi_i + S_\Phi$$

where the coefficients A_i (i=E,W,N,S,T,B) contain both the convective and diffusive fluxes. these equations are assembled and solved by calling subroutine SOLSIP which is based on Stone's Strongly Implicit Solver [4].

Subroutine SOLSIP

This subroutine solves the system of linear algebraic equations for k and ε using Stone's Implicit Procedure [4]. The array RES (IJK) is used to store the residuals. The sum of absolute residuals "RES1" calculated in the first pass through this part of the routine is used as a measure of convergence of the solution process as a whole and this value is stored in RESOR (IPHI). This variable RESOR (IPHI) is passed to the main solver and if desired can be normalized and compared with the maximum error allowed there. If necessary inner iterations counter L and the sum of absolute residuals RES1 are printed out to monitor the rate of convergence of k and ε solution. If the ratio RSM is greater than the maximum allowed for the variable in question, SOR (IPHI), and the number of inner iterations is smaller than a prescribed maximum, NSWP (IPHI), then the routine repeats the sequence of calculating the residuals, increment vectors and updating the dependent variable.

Subroutine MODVIS

This section calculates the effective viscosity and is called after calculating k and ε. At locations where ε is close to zero (i.e., $\leq 10^{-30}$) viscosity is set to zero. A provision is made for under relaxing changes in effective viscosity which may help to stabilize oscillations and improve convergence rate.

Subroutines MODK and MODED

These subroutines are called from subroutine CALCE and they set the boundary conditions for k and ε. For the kinetic energy equation for example, the bottom boundary is checked first for one of the options below;

(1) An inflow boundary BCFB(IJ) = 1 (IJ = (I-1)*NJ+J), where the source term is set to accept the inlet values at the x-y plane (bottom boundary K=1).

(2) Outflow boundary BCFB(IJ) = 2, where zero gradient in the z-direction is employed.

(3) Symmetry boundary, BCFB(IJ) = 3, where gradients normal to symmetry x-y plane are zero.

(4) Wall boundary, BCFB(IJ) = 4, where the turbulent kinetic energy production (per unit volume) term GENTB(I) calculated form subroutine WALLFN in program MODIFY is added to the rest of the source term SU(IJK).

Boundary conditions for the ε-equation are similar to those of k except at the wall where they are set to appropriate values for each near wall treatment.

Program MODIFY

This program is compiled separately and is called from the u, v and w momentum solver .It basically updates the flux source term of the discretized momentum equation due to wall shear stresses. If the u-momentum equation for example is discretized in the form

$$A_p^* u_p = \sum_{i=EWNSTB} A_i u_i + S_u^*$$

where P, E, W, N, S, T, B are cell nodes, and A_p^* and A_i's contain convective and diffusive coefficients. S_u^* is the source term containing pressure gradients and cross-derivative diffusion terms and convective terms for second-order upwinding scheme. This source term is usually linearized as $S_u^* = S_u - B_p u_p$ The term B_p is usually moved to the left hand side of the equation and modifies the diagonal coefficient $A_p = A_p^* + B_p$, and the equation can be written as

$$A_p u_p = \sum_{i=EWNSTB} A_i u_i + S_u$$

Then S_u and B_p are passed to subroutine MODIFY where they are modified if a wall is present (e.g., BCFB(IJ) = 4 for bottom boundary).

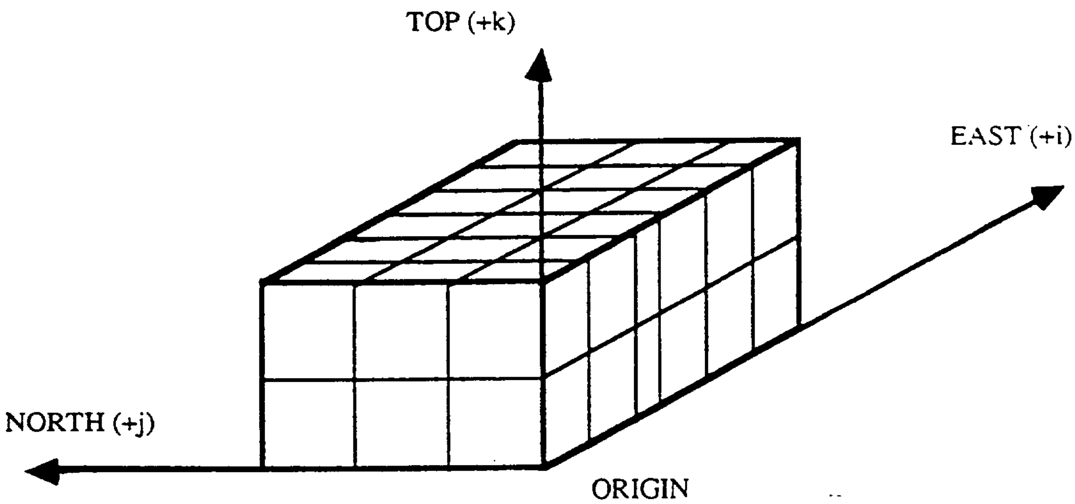


Figure 3. Cell volume and coordinate system

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C     3D-SINGLE-SCALE K-E TURBULENCE MODULE                          C
C                                                                    C
C     Rocketdyne CFD Technology Center                               C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C----------------------------------------
      SUBROUTINE KEMOD3 (NIM,NJM,NKM,
     & X,Y,Z,U,V,W,TE,ED,
     & URFK,URFE,PRTK,PRTE,G,F1,F2,F3,ITER,
     & VISCOS,VIS,C1,C2,CMU,
     & BCFE,BCFW,BCFS,BCFN,BCFB,BCFT)
C
C----------------------------------------
      INCLUDE 'kemod.h'
C
      DIMENSION  LK(JZ),LI(JX),
     &           X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),
     &           FX(JXYZ),FY(JXYZ),FZ(JXYZ)
      DIMENSION  U(JXYZ),V(JXYZ),W(JXYZ),
     &           TE(JXYZ),ED(JXYZ),VIS(JXYZ)
C
C---- CALCULATE GRID GEOMETRIC VARIABLES INITIALLY
C
      IF(ITER.LE.1) THEN
         CALL GRIDG
      ENDIF
C
C-- CALL KINETIC ENERGY SOLVER
      CALL CALCE (TE,1)
C-- CALL ENERGY DISSIPATION SOLVER
      CALL CALCE (ED,2)
C
C-- UPDATE AND CALCULATE THE EDDY VISCOSITY
      CALL MODVIS
C
C
      RETURN
      END
C
C------------------------------------------------
      SUBROUTINE GRIDG
      INCLUDE 'kemod.h'
C
      NI = NIM + 1
      NJ = NJM + 1
      NK = NKM + 1
C
      DO 10 I=1,NI
      LI(I)=(I-1)*NJ
10    CONTINUE
      DO 20 K=1,NK
      LK(K)=(K-1)*NI*NJ
20    CONTINUE
C
       DO 25 K=1,NK
       DO 25 J=1,NJ
       IL=LI(NI)+LK(K)+J
       ILM=IL-NJ
       X(IL)=X(ILM)
       Y(IL)=Y(ILM)
       Z(IL)=Z(ILM)
25     CONTINUE
C
       DO 40 I=1,NI
       DO 40 J=1,NJ
       JL=J+LI(I)+LK(NK)
       JLMK=JL-NI*NJ
       X(JL)=X(JLMK)
```



```
       Y(JL)=Y(JLMK)
       Z(JL)=Z(JLMK)
40     CONTINUE
C$END
       DO 30 I=1,NI
       DO 30 K=1,NK
       JK=NJ+LI(I)+LK(K)
       JKM=JK-1
       X(JK)=X(JKM)
       Y(JK)=Y(JKM)
       Z(JK)=Z(JKM)
30     CONTINUE
C
      NIM=NI-1
      NJM=NJ-1
      NKM=NK-1
      NIMM=NIM-1
      NJMM=NJM-1
      NKMM=NKM-1
      NIJ=NI*NJ
      NIJK=NIJ*NK
C
      VOL1(A1,A2,A3,B1,B2,B3,Q1,Q2,Q3)=(A2*B3-B2*A3)*Q1+(B1*A3-A1*B3)*Q2
     *     +(A1*B2-A2*B1)*Q3
C
C-----CALCULATION OF CELL VOLUMES
C
      DO 350 IJK=1,NIJK
  350 VOL(IJK)=0.
      SIXR=1./6.
      DO 400 K=2,NKM
      LKK=LK(K)
      DO 400 I=2,NIM
      LIK=LKK+LI(I)
      DO 400 J=2,NJM
      IJK=LIK+J
      IMJK=IJK-NJ
      IMJMK=IMJK-1
      IJMK=IJK-1
      IJKM=IJK-NIJ
      IJKA=IJKM-NJ-1
C
      XA=X(IJKA)
      YA=Y(IJKA)
      ZA=Z(IJKA)
      DXAD=X(IJMK)-XA
      DYAD=Y(IJMK)-YA
      DZAD=Z(IJMK)-ZA
      DXAB=X(IJKM-1)-XA
      DYAB=Y(IJKM-1)-YA
      DZAB=Z(IJKM-1)-ZA
      DXAC=X(IMJMK)-XA
      DYAC=Y(IMJMK)-YA
      DZAC=Z(IMJMK)-ZA
      DXAE=X(IJKM-NJ)-XA
      DYAE=Y(IJKM-NJ)-YA
      DZAE=Z(IJKM-NJ)-ZA
      DXAF=X(IJKM)-XA
      DYAF=Y(IJKM)-YA
      DZAF=Z(IJKM)-ZA
      DXAG=X(IMJK)-XA
      DYAG=Y(IMJK)-YA
      DZAG=Z(IMJK)-ZA
      DXAH=X(IJK)-XA
      DYAH=Y(IJK)-YA
      DZAH=Z(IJK)-ZA
C
      VOLUM=VOL1(DXAD,DYAD,DZAD,DXAB,DYAB,DZAB,DXAH,DYAH,DZAH)+
     *      VOL1(DXAC,DYAC,DZAC,DXAD,DYAD,DZAD,DXAH,DYAH,DZAH)+
     *      VOL1(DXAG,DYAG,DZAG,DXAC,DYAC,DZAC,DXAH,DYAH,DZAH)+
     *      VOL1(DXAB,DYAB,DZAB,DXAF,DYAF,DZAF,DXAH,DYAH,DZAH)+
```

```
     *         VOL1(DXAF,DYAF,DZAF,DXAE,DYAE,DZAE,DXAH,DYAH,DZAH)+
     *         VOL1(DXAE,DYAE,DZAE,DXAG,DYAG,DZAG,DXAH,DYAH,DZAH)
      VOL(IJK)=VOLUM*SIXR
  400 CONTINUE
C
C-----CALCULATIN OF INTERPOLATION FACTORS
C
      DO 450 IJK=1,NIJK
      FX(IJK)=0.
      FY(IJK)=0.
  450 FZ(IJK)=0.
C------KSI - DIRECTION
      DO 500 K=2,NKM
      LKK=LK(K)
      DO 500 J=2,NJM
      DO 500 I=2,NIM
      IJK=LKK+LI(I)+J
      IJKM=IJK-NIJ
      IPJK=IJK+NJ
      IMJK=IJK-NJ
      IPJKM=IJKM+NJ
      IMJKM=IJKM-NJ
      DXKSP=QTR*(X(IJK)+X(IJK-1)+X(IJKM)+X(IJKM-1)-
     *             X(IMJK)-X(IMJK-1)-X(IMJKM)-X(IMJKM-1))
      DXKSE=QTR*(X(IPJK)+X(IPJK-1)+X(IPJKM)+X(IPJKM-1)-
     *             X(IJK)-X(IJK-1)-X(IJKM)-X(IJKM-1))
      DYKSP=QTR*(Y(IJK)+Y(IJK-1)+Y(IJKM)+Y(IJKM-1)-
     *             Y(IMJK)-Y(IMJK-1)-Y(IMJKM)-Y(IMJKM-1))
      DYKSE=QTR*(Y(IPJK)+Y(IPJK-1)+Y(IPJKM)+Y(IPJKM-1)-
     *             Y(IJK)-Y(IJK-1)-Y(IJKM)-Y(IJKM-1))
      DZKSP=QTR*(Z(IJK)+Z(IJK-1)+Z(IJKM)+Z(IJKM-1)-
     *             Z(IMJK)-Z(IMJK-1)-Z(IMJKM)-Z(IMJKM-1))
      DZKSE=QTR*(Z(IPJK)+Z(IPJK-1)+Z(IPJKM)+Z(IPJKM-1)-
     *             Z(IJK)-Z(IJK-1)-Z(IJKM)-Z(IJKM-1))
      DKSP=SQRT(DXKSP**2+DYKSP**2+DZKSP**2)
      DKSE=SQRT(DXKSE**2+DYKSE**2+DZKSE**2)
      FX(IJK)=DKSP/(DKSP+DKSE)
  500 CONTINUE
      DO 520 I=1,NIM
      LII=LI(I)
      DO 520 J=2,NJM
      IJK=LII+J
      FX(IJK)=FX(IJK+NIJ)
      IJK=LK(NK)+IJK
  520 FX(IJK)=FX(IJK-NIJ)
      DO 530 K=1,NK
      LKK=LK(K)
      DO 530 I=1,NIM
      IJK=LKK+LI(I)+1
      FX(IJK)=FX(IJK+1)
      IJK=IJK+NJM
  530 FX(IJK)=FX(IJK-1)
C------ETA - DIRECTION
      DO 600 K=2,NKM
      LKK=LK(K)
      DO 600 I=2,NIM
      LII=LI(I)
      DO 600 J=2,NJM
      IJK=LKK+LII+J
      IMJK=IJK-NJ
      IJKM=IJK-NIJ
      IMJKM=IJKM-NJ
      DXETP=QTR*(X(IJK)+X(IMJK)+X(IJKM)+X(IMJKM)-
     *             X(IJK-1)-X(IMJK-1)-X(IJKM-1)-X(IMJKM-1))
      DXETN=QTR*(X(IJK+1)+X(IMJK+1)+X(IJKM+1)+X(IMJKM+1)-
     *             X(IJK)-X(IMJK)-X(IJKM)-X(IMJKM))
      DYETP=QTR*(Y(IJK)+Y(IMJK)+Y(IJKM)+Y(IMJKM)-
     *             Y(IJK-1)-Y(IMJK-1)-Y(IJKM-1)-Y(IMJKM-1))
      DYETN=QTR*(Y(IJK+1)+Y(IMJK+1)+Y(IJKM+1)+Y(IMJKM+1)-
     *             Y(IJK)-Y(IMJK)-Y(IJKM)-Y(IMJKM))
      DZETP=QTR*(Z(IJK)+Z(IMJK)+Z(IJKM)+Z(IMJKM)-
```



```
     *             Z(IJK-1)-Z(IMJK-1)-Z(IJKM-1)-Z(IMJKM-1))
      DZETN=QTR*(Z(IJK+1)+Z(IMJK+1)+Z(IJKM+1)+Z(IMJKM+1)-
     *             Z(IJK)-Z(IMJK)-Z(IJKM)-Z(IMJKM))
      DETP=SQRT(DXETP**2+DYETP**2+DZETP**2)
      DETN=SQRT(DXETN**2+DYETN**2+DZETN**2)
      FY(IJK)=DETP/(DETP+DETN)
  600 CONTINUE
      DO 620 I=2,NIM
      LII=LI(I)
      DO 620 J=1,NJM
      IJK=LII+J
      FY(IJK)=FY(IJK+NIJ)
      IJK=LK(NK)+IJK
  620 FY(IJK)=FY(IJK-NIJ)
      DO 630 K=1,NK
      LKK=LK(K)
      DO 630 J=1,NJM
      IJK=LKK+J
      FY(IJK)=FY(IJK+NJ)
      IJK=LI(NI)+IJK
  630 FY(IJK)=FY(IJK-NJ)
C------ZETA - DIRECTION
      DO 700 I=2,NIM
      LII=LI(I)
      DO 700 J=2,NJM
      DO 700 K=2,NKM
      IJK=LK(K)+LII+J
      IJKM=IJK-NIJ
      IJKP=IJK+NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      IMJKP=IJKP-NJ
      DXZDP=QTR*(X(IJK)+X(IMJK)+X(IJK-1)+X(IMJK-1)-
     *             X(IJKM)-X(IMJKM)-X(IJKM-1)-X(IMJKM-1))
      DXZDT=QTR*(X(IJKP)+X(IJKP-1)+X(IMJKP)+X(IMJKP-1)-
     *             X(IJK)-X(IJK-1)-X(IMJK)-X(IMJK-1))
      DYZDP=QTR*(Y(IJK)+Y(IMJK)+Y(IJK-1)+Y(IMJK-1)-
     *             Y(IJKM)-Y(IMJKM)-Y(IJKM-1)-Y(IMJKM-1))
      DYZDT=QTR*(Y(IJKP)+Y(IJKP-1)+Y(IMJKP)+Y(IMJKP-1)-
     *             Y(IJK)-Y(IJK-1)-Y(IMJK)-Y(IMJK-1))
      DZZDP=QTR*(Z(IJK)+Z(IMJK)+Z(IJK-1)+Z(IMJK-1)-
     *             Z(IJKM)-Z(IMJKM)-Z(IJKM-1)-Z(IMJKM-1))
      DZZDT=QTR*(Z(IJKP)+Z(IJKP-1)+Z(IMJKP)+Z(IMJKP-1)-
     *             Z(IJK)-Z(IJK-1)-Z(IMJK)-Z(IMJK-1))
      DZDP=SQRT(DXZDP**2+DYZDP**2+DZZDP**2)
      DZDT=SQRT(DXZDT**2+DYZDT**2+DZZDT**2)
      FZ(IJK)=DZDP/(DZDP+DZDT)
  700 CONTINUE
      DO 720 K=1,NKM
      LKK=LK(K)
      DO 720 J=2,NJM
      IJK=LKK+J
      FZ(IJK)=FZ(IJK+NJ)
      IJK=LI(NI)+IJK
  720 FZ(IJK)=FZ(IJK-NJ)
      DO 730 K=1,NKM
      LKK=LK(K)
      DO 730 I=1,NI
      IJK=LKK+LI(I)+1
      FZ(IJK)=FZ(IJK+1)
      IJK=IJK+NJM
  730 FZ(IJK)=FZ(IJK-1)
C
C-----NORMAL DISTANCES FROM DOMAIN BOUNDARIES
C
      DO 740 IJ=1,NIJ
      DNB(IJ)=0.
  740 DNT(IJ)=0.
      NIK=NI*NK
      NJK=NJ*NK
      DO 741 IK=1,NIK
```

```
      DNS(IK)=0.
  741 DNN(IK)=0.
      DO 742 JK=1,NJK
      DNW(JK)=0.
  742 DNE(JK)=0.
      DO 750 I=2,NIM
      LII=LI(I)
      DO 760 J=2,NJM
C------BOTTOM  BOUNDARY
      IJ=LII+J
      IJK=LK(2)+IJ
      IJKM=IJK-NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      DX1=X(IJKM-1)-X(IMJKM)
      DY1=Y(IJKM-1)-Y(IMJKM)
      DZ1=Z(IJKM-1)-Z(IMJKM)
      DX2=X(IJKM)-X(IMJKM-1)
      DY2=Y(IJKM)-Y(IMJKM-1)
      DZ2=Z(IJKM)-Z(IMJKM-1)
      XB=QTR*(X(IJKM)+X(IJKM-1)+X(IMJKM)+X(IMJKM-1))
      YB=QTR*(Y(IJKM)+Y(IJKM-1)+Y(IMJKM)+Y(IMJKM-1))
      ZB=QTR*(Z(IJKM)+Z(IJKM-1)+Z(IMJKM)+Z(IMJKM-1))
      XP=HAF*(XB+QTR*(X(IJK)+X(IJK-1)+X(IMJK)+X(IMJK-1)))
      YP=HAF*(YB+QTR*(Y(IJK)+Y(IJK-1)+Y(IMJK)+Y(IMJK-1)))
      ZP=HAF*(ZB+QTR*(Z(IJK)+Z(IJK-1)+Z(IMJK)+Z(IMJK-1)))
      DX3=XP-XB
      DY3=YP-YB
      DZ3=ZP-ZB
      DNB(IJ)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
C------TOP  BOUNDARY
      IJK=LK(NKM)+IJ
      IJKM=IJK-NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      DX1=X(IJK)-X(IMJK-1)
      DY1=Y(IJK)-Y(IMJK-1)
      DZ1=Z(IJK)-Z(IMJK-1)
      DX2=X(IJK-1)-X(IMJK)
      DY2=Y(IJK-1)-Y(IMJK)
      DZ2=Z(IJK-1)-Z(IMJK)
      XB=QTR*(X(IJK)+X(IJK-1)+X(IMJK)+X(IMJK-1))
      YB=QTR*(Y(IJK)+Y(IJK-1)+Y(IMJK)+Y(IMJK-1))
      ZB=QTR*(Z(IJK)+Z(IJK-1)+Z(IMJK)+Z(IMJK-1))
      XP=HAF*(XB+QTR*(X(IJKM)+X(IJKM-1)+X(IMJKM)+X(IMJKM-1)))
      YP=HAF*(YB+QTR*(Y(IJKM)+Y(IJKM-1)+Y(IMJKM)+Y(IMJKM-1)))
      ZP=HAF*(ZB+QTR*(Z(IJKM)+Z(IJKM-1)+Z(IMJKM)+Z(IMJKM-1)))
      DX3=XP-XB
      DY3=YP-YB
      DZ3=ZP-ZB
      DNT(IJ)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
  760 CONTINUE
C------SOUTH  BOUNDARY
      DO 770 K=2,NKM
      IK=(I-1)*NK+K
      IJK=LII+LK(K)+2
      IJKM=IJK-NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      DX1=X(IMJK-1)-X(IJKM-1)
      DY1=Y(IMJK-1)-Y(IJKM-1)
      DZ1=Z(IMJK-1)-Z(IJKM-1)
      DX2=X(IJK-1)-X(IMJKM-1)
      DY2=Y(IJK-1)-Y(IMJKM-1)
      DZ2=Z(IJK-1)-Z(IMJKM-1)
      XB=QTR*(X(IJK-1)+X(IMJK-1)+X(IJKM-1)+X(IMJKM-1))
      YB=QTR*(Y(IJK-1)+Y(IMJK-1)+Y(IJKM-1)+Y(IMJKM-1))
      ZB=QTR*(Z(IJK-1)+Z(IMJK-1)+Z(IJKM-1)+Z(IMJKM-1))
      XP=HAF*(XB+QTR*(X(IJK)+X(IMJK)+X(IJKM)+X(IMJKM)))
      YP=HAF*(YB+QTR*(Y(IJK)+Y(IMJK)+Y(IJKM)+Y(IMJKM)))
      ZP=HAF*(ZB+QTR*(Z(IJK)+Z(IMJK)+Z(IJKM)+Z(IMJKM)))
```



```
      DX3=XP-XB
      DY3=YP-YB
      DZ3=ZP-ZB
      DNS(IK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
C------NORTH  BOUNDARY
      IJK=LK(K)+LII+NJM
      IJKM=IJK-NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      DX1=X(IJK)-X(IMJKM)
      DY1=Y(IJK)-Y(IMJKM)
      DZ1=Z(IJK)-Z(IMJKM)
      DX2=X(IMJK)-X(IJKM)
      DY2=Y(IMJK)-Y(IJKM)
      DZ2=Z(IMJK)-Z(IJKM)
      XB=QTR*(X(IJK)+X(IMJK)+X(IJKM)+X(IMJKM))
      YB=QTR*(Y(IJK)+Y(IMJK)+Y(IJKM)+Y(IMJKM))
      ZB=QTR*(Z(IJK)+Z(IMJK)+Z(IJKM)+Z(IMJKM))
      XP=HAF*(XB+QTR*(X(IJK-1)+X(IMJK-1)+X(IJKM-1)+X(IMJKM-1)))
      YP=HAF*(YB+QTR*(Y(IJK-1)+Y(IMJK-1)+Y(IJKM-1)+Y(IMJKM-1)))
      ZP=HAF*(ZB+QTR*(Z(IJK-1)+Z(IMJK-1)+Z(IJKM-1)+Z(IMJKM-1)))
      DX3=XP-XB
      DY3=YP-YB
      DZ3=ZP-ZB
      DNN(IK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
  770 CONTINUE
  750 CONTINUE
C------WEST  BOUNDARY
      DO 780 K=2,NKM
      LKK=LK(K)
      DO 780 J=2,NJM
      JK=(K-1)*NJ+J
      IJK=LKK+LI(2)+J
      IJKM=IJK-NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      DX1=X(IMJK)-X(IMJKM-1)
      DY1=Y(IMJK)-Y(IMJKM-1)
      DZ1=Z(IMJK)-Z(IMJKM-1)
      DX2=X(IMJK-1)-X(IMJKM)
      DY2=Y(IMJK-1)-Y(IMJKM)
      DZ2=Z(IMJK-1)-Z(IMJKM)
      XB=QTR*(X(IMJK)+X(IMJK-1)+X(IMJKM)+X(IMJKM-1))
      YB=QTR*(Y(IMJK)+Y(IMJK-1)+Y(IMJKM)+Y(IMJKM-1))
      ZB=QTR*(Z(IMJK)+Z(IMJK-1)+Z(IMJKM)+Z(IMJKM-1))
      XP=HAF*(XB+QTR*(X(IJK)+X(IJK-1)+X(IJKM)+X(IJKM-1)))
      YP=HAF*(YB+QTR*(Y(IJK)+Y(IJK-1)+Y(IJKM)+Y(IJKM-1)))
      ZP=HAF*(ZB+QTR*(Z(IJK)+Z(IJK-1)+Z(IJKM)+Z(IJKM-1)))
      DX3=XP-XB
      DY3=YP-YB
      DZ3=ZP-ZB
      DNW(JK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
C------EAST  BOUNDARY
      IJK=LKK+LI(NIM)+J
      IJKM=IJK-NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      DX1=X(IJK-1)-X(IJKM)
      DY1=Y(IJK-1)-Y(IJKM)
      DZ1=Z(IJK-1)-Z(IJKM)
      DX2=X(IJK)-X(IJKM-1)
      DY2=Y(IJK)-Y(IJKM-1)
      DZ2=Z(IJK)-Z(IJKM-1)
      XB=QTR*(X(IJK)+X(IJK-1)+X(IJKM)+X(IJKM-1))
      YB=QTR*(Y(IJK)+Y(IJK-1)+Y(IJKM)+Y(IJKM-1))
      ZB=QTR*(Z(IJK)+Z(IJK-1)+Z(IJKM)+Z(IJKM-1))
      XP=HAF*(XB+QTR*(X(IMJK)+X(IMJK-1)+X(IMJKM)+X(IMJKM-1)))
      YP=HAF*(YB+QTR*(Y(IMJK)+Y(IMJK-1)+Y(IMJKM)+Y(IMJKM-1)))
      ZP=HAF*(ZB+QTR*(Z(IMJK)+Z(IMJK-1)+Z(IMJKM)+Z(IMJKM-1)))
      DX3=XP-XB
      DY3=YP-YB
```

```
      DZ3=ZP-ZB
      DNE(JK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
  780 CONTINUE
C
C  CALCULATE THE NORMAL DISTANCES FROM BOUNDARIES
C
      FUNCTION DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
      XAN=DY1*DZ2-DY2*DZ1
      YAN=DX2*DZ1-DX1*DZ2
      ZAN=DX1*DY2-DX2*DY1
      ARNR=1./SQRT(XAN**2+YAN**2+ZAN**2)
      DELN=(DX3*XAN+DY3*YAN+DZ3*ZAN)*ARNR
      RETURN
      END
C
C-----------------------------------------------------------
      SUBROUTINE CALCE (PHI.IPHI)
      INCLUDE 'kemod.h'
C
      DIMENSION PHIWB(JY),PHI(JXYZ)
      INTEGER BCFW,BCFE,BCFS,BCFN,BCDB,BCFT
C
C-----INITIALISE TEMPORARILY STORED VARIABLES
C
      G=GKE
      IF(IPHI.EQ.1) THEN
      URFPHI=1./URFK
      PRTINVP=1./PRTK
      ELSE
      URFPHI=1./URFE
      PRTINVP=1./PRTE
      ENDIF
C
      DO 5 J=1,NJ
    5 PHIWB(J)=PHI(J)
C-----BOTTOM BOUNDARY CELL FACES
      DO 10 I=2,NIM
      LII=LI(I)
      PHIS=PHI(LII+1)
      DO 10 J=2,NJM
      IJ=LII+J
      IJK=IJ
      IJKP=IJK+NIJ
      IMJK=IJK-NJ
      IJMK=IJK-1
      IMJMK=IMJK-1
      VOLT=HAF*(VOL(IJK)+VOL(IJKP))
      DXKS=HAF*(X(IJK)-X(IMJK)+X(IJMK)-X(IMJMK))
      DXET=HAF*(X(IJK)-X(IJMK)+X(IMJK)-X(IMJMK))
      DXZD=AHT*(X(IJKP)+X(IJKP-1)+X(IJKP-NJ)+X(IJKP-NJ-1)-
     &          XZB(IJ)-XZB(IJ-1)-XZB(IJ-NJ)-XZB(IJ-NJ-1) )
      DYKS=HAF*(Y(IJK)-Y(IMJK)+Y(IJMK)-Y(IMJMK))
      DYET=HAF*(Y(IJK)-Y(IJMK)+Y(IMJK)-Y(IMJMK))
      DYZD=AHT*(Y(IJKP)+Y(IJKP-1)+Y(IJKP-NJ)+Y(IJKP-NJ-1)-
     &          YZB(IJ)-YZB(IJ-1)-YZB(IJ-NJ)-YZB(IJ-NJ-1) )
      DZKS=HAF*(Z(IJK)-Z(IMJK)+Z(IJMK)-Z(IMJMK))
      DZET=HAF*(Z(IJK)-Z(IJMK)+Z(IMJK)-Z(IMJMK))
      DZZD=AHT*(Z(IJKP)+Z(IJKP-1)+Z(IJKP-NJ)+Z(IJKP-NJ-1)-
     &          ZZB(IJ)-ZZB(IJ-1)-ZZB(IJ-NJ)-ZZB(IJ-NJ-1) )
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
      ART=B31**2+B32**2+B33**2
```



```
      GAMT=VIS(IJK)*PRTINVP/VOLT
      DB(IJ)=GAMT*ART
      FZBB(IJ)=1.0
      PHIE=PHI(IJK)*(1.-FX(IJK))+PHI(IJK+NJ)*FX(IJK)
      PHIN=PHI(IJK)*(1.-FY(IJK))+PHI(IJK+1)*FY(IJK)
      SUB(IJ)=GAMT*((B11*B31+B12*B32+B13*B33)*(PHIE-PHIWB(J))+
     *               (B21*B31+B22*B32+B23*B33)*(PHIN-PHIS))
      PHIS=PHIN
      PHIWB(J)=PHIE
   10 CONTINUE
C-----WEST BOUNDARY CELL FACES
      DO 100 K=2,NKM
      LKK=LK(K)
      KKJ=(K-1)*NJ
      KKI=(K-1)*NI
      PHIS=PHI(LKK+1)
      DO 20 J=2,NJM
      JK=KKJ+J
      IJK=LKK+J
      IJMK=IJK-1
      IJKM=IJK-NIJ
      IJMKM=IJKM-1
      VOLE=HAF*(VOL(IJK)+VOL(IJK+NJ))
      DXKS=AHT*(X(IJK+NJ)+X(IJMK+NJ)+X(IJKM+NJ)+X(IJMKM+NJ)-
     &          XZW(JK)-XZW(JK-1)-XZW(JK-NJ)-XZW(JK-NJ-1) )
      DXET=HAF*(X(IJK)-X(IJMK)+X(IJKM)-X(IJMKM))
      DXZD=HAF*(X(IJK)-X(IJKM)+X(IJMK)-X(IJMKM))
      DYKS=AHT*(Y(IJK+NJ)+Y(IJMK+NJ)+Y(IJKM+NJ)+Y(IJMKM+NJ)-
     &          YZW(JK)-YZW(JK-1)-YZW(JK-NJ)-YZW(JK-NJ-1) )
      DYET=HAF*(Y(IJK)-Y(IJMK)+Y(IJKM)-Y(IJMKM))
      DYZD=HAF*(Y(IJK)-Y(IJKM)+Y(IJMK)-Y(IJMKM))
      DZKS=AHT*(Z(IJK+NJ)+Z(IJMK+NJ)+Z(IJKM+NJ)+Z(IJMKM+NJ)-
     &          ZZW(JK)-ZZW(JK-1)-ZZW(JK-NJ)-ZZW(JK-NJ-1) )
      DZET=HAF*(Z(IJK)-Z(IJMK)+Z(IJKM)-Z(IJMKM))
      DZZD=HAF*(Z(IJK)-Z(IJKM)+Z(IJMK)-Z(IJMKM))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
      ARE=B11**2+B12**2+B13**2
      GAME=VIS(IJK)*PRTINVP/VOLE
      DW(J)=GAME*ARE
      FXWW(J)=1.0
      PHIN=PHI(IJK)*(1.-FY(IJK))+PHI(IJK+1)*FY(IJK)
      PHIT=PHI(IJK)*(1.-FZ(IJK))+PHI(IJK+NIJ)*FZ(IJK)
      PHIB=PHI(IJKM)*(1.-FZ(IJKM))+PHI(IJK)*FZ(IJKM)
      SUW(J)=GAME*((B21*B11+B22*B12+B23*B13)*(PHIN-PHIS)+
     *             (B31*B11+B32*B12+B33*B13)*(PHIT-PHIB))
      PHIS=PHIN
   20 CONTINUE
C-----SOUTH BOUNDARY CELL FACES
      DO 100 I=2,NIM
      IK=(I-1)*NK+K
      LII=LI(I)
      IJK=LKK+LII+1
      IJKM=IJK-NIJ
      IMJK=IJK-NJ
      IMJKM=IJKM-NJ
      VOLN=HAF*(VOL(IJK)+VOL(IJK+1))
      DXKS=HAF*(X(IJK)-X(IMJK)+X(IJKM)-X(IMJKM))
      DXET=AHT*(X(IJK+1)+X(IMJK+1)+X(IJKM+1)+X(IMJKM+1)-
     &          XZS(IK)-XZS(IK-1)-XZS(IK-NK)-XZS(IK-NK-1) )
      DXZD=HAF*(X(IJK)-X(IJKM)+X(IMJK)-X(IMJKM))
      DYKS=HAF*(Y(IJK)-Y(IMJK)+Y(IJKM)-Y(IMJKM))
```

```
      DYET=AHT*(Y(IJK+1)+Y(IMJK+1)+Y(IJKM+1)+Y(IMJKM+1)-
     &          YZS(IK)-YZS(IK-1)-YZS(IK-NK)-YZS(IK-NK-1) )
      DYZD=HAF*(Y(IJK)-Y(IJKM)+Y(IMJK)-Y(IMJKM))
      DZKS=HAF*(Z(IJK)-Z(IMJK)+Z(IJKM)-Z(IMJKM))
      DZET=AHT*(Z(IJK+1)+Z(IMJK+1)+Z(IJKM+1)+Z(IMJKM+1)-
     &          ZZS(IK)-ZZS(IK-1)-ZZS(IK-NK)-ZZS(IK-NK-1) )
      DZZD=HAF*(Z(IJK)-Z(IJKM)+Z(IMJK)-Z(IMJKM))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
      ARN=B21**2+B22**2+B23**2
      GAMN=VIS(IJK)*PRTINVP/VOLN
      DN=GAMN*ARN
      FYSS=1.0
      PHIE=PHI(IJK)*(1.-FX(IJK))+PHI(IJK+NJ)*FX(IJK)
      PHIW=PHI(IMJK)*(1.-FX(IMJK))+PHI(IJK)*FX(IMJK)
      PHIT=PHI(IJK)*(1.-FZ(IJK))+PHI(IJK+NIJ)*FZ(IJK)
      PHIB=PHI(IJKM)*(1.-FZ(IJKM))+PHI(IJK)*FZ(IJKM)
      SUN=GAMN*((B11*B21+B12*B22+B13*B23)*(PHIE-PHIW)+
     *          (B31*B21+B32*B22+B33*B23)*(PHIT-PHIB))
C
C-----THE MAIN LOOP
C
      DO 100 J=2,NJM
      IJ=LII+J
      IJK=LKK+IJ
      IJKP=IJK+NIJ
      IJKM=IJK-NIJ
      IPJK=IJK+NJ
      IMJK=IJK-NJ
      IJPK=IJK+1
      IJMK=IJK-1
      FXE=FX(IJK)
      FXW=1.-FXE
      FYN=FY(IJK)
      FYS=1.-FYN
      FZT=FZ(IJK)
      FZB=1.-FZT
C
      VOLE=HAF*(VOL(IJK)+VOL(IPJK))
      VOLN=HAF*(VOL(IJK)+VOL(IJPK))
      VOLT=HAF*(VOL(IJK)+VOL(IJKP))
      GAME=(VIS(IJK)*FXW+VIS(IPJK)*FXE)*PRTINVP/VOLE
      GAMN=(VIS(IJK)*FYS+VIS(IJPK)*FYN)*PRTINVP/VOLN
      GAMT=(VIS(IJK)*FZB+VIS(IJKP)*FZT)*PRTINVP/VOLT
C
C-----EAST CELL FACE
C
      DXKS=AHT*(X(IPJK)+X(IPJK-1)+X(IJKM+NJ)+X(IJKM+NJM)-
     *            X(IMJK)-X(IMJK-1)-X(IJKM-NJ)-X(IJKM-NJ-1))
      DXET=HAF*(X(IJK)-X(IJMK)+X(IJKM)-X(IJKM-1))
      DXZD=HAF*(X(IJK)-X(IJKM)+X(IJMK)-X(IJKM-1))
      DYKS=AHT*(Y(IPJK)+Y(IPJK-1)+Y(IJKM+NJ)+Y(IJKM+NJM)-
     *            Y(IMJK)-Y(IMJK-1)-Y(IJKM-NJ)-Y(IJKM-NJ-1))
      DYET=HAF*(Y(IJK)-Y(IJMK)+Y(IJKM)-Y(IJKM-1))
      DYZD=HAF*(Y(IJK)-Y(IJKM)+Y(IJMK)-Y(IJKM-1))
      DZKS=AHT*(Z(IPJK)+Z(IPJK-1)+Z(IJKM+NJ)+Z(IJKM+NJM)-
     *            Z(IMJK)-Z(IMJK-1)-Z(IJKM-NJ)-Z(IJKM-NJ-1))
      DZET=HAF*(Z(IJK)-Z(IJMK)+Z(IJKM)-Z(IJKM-1))
      DZZD=HAF*(Z(IJK)-Z(IJKM)+Z(IJMK)-Z(IJKM-1))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
```



```
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
      ARE=B11**2+B12**2+B13**2
      DE=GAME*ARE
      AEE=AMIN1(F1(IJK),0.)*FX(IPJK)*G
      AWW=-AMAX1(F1(IMJK),0.)*(1.-FXWW(J))*G
      AE1=-AMIN1(F1(IMJK),0.)*FXE*G
      AE(IJK)=DE-AMIN1(F1(IJK),0.)-AEE
      AW1=AMAX1(F1(IJK),0.)*(1.-FX(IMJK))*G
      AW(IJK)=DW(J)+AMAX1(F1(IMJK),0.)-AWW
C
      PHIE=PHI(IJK)*FXW+PHI(IPJK)*FXE
      PHIBE=(PHI(IJKM)*FXW+PHI(IJKM+NJ)*FXE)*(1.-FZ(IJKM))+PHIE*FZ(IJKM)
      PHITE=(PHI(IJKP)*FXW+PHI(IJKP+NJ)*FXE)*FZT+PHIE*FZB
      PHISE=(PHI(IJMK)*FXW+PHI(IPJK-1)*FXE)*(1.-FY(IJMK))+PHIE*FY(IJMK)
      PHINE=(PHI(IJPK)*FXW+PHI(IPJK+1)*FXE)*FYN+PHIE*FYS
      SUE=GAME*((B11*B21+B12*B22+B13*B23)*(PHINE-PHISE)+
     *          (B11*B31+B12*B32+B13*B33)*(PHITE-PHIBE))
C
C-----NORTH  CELL - FACE
C
      DXKS=HAF*(X(IJK)-X(IMJK)+X(IJKM)-X(IJKM-NJ))
      DXET=AHT*(X(IJPK)+X(IMJK+1)+X(IJKM+1)+X(IJKM-NJM)-
     *            X(IJMK)-X(IMJK-1)-X(IJKM-1)-X(IJKM-NJ-1))
      DXZD=HAF*(X(IJK)-X(IJKM)+X(IMJK)-X(IJKM-NJ))
      DYKS=HAF*(Y(IJK)-Y(IMJK)+Y(IJKM)-Y(IJKM-NJ))
      DYET=AHT*(Y(IJPK)+Y(IMJK+1)+Y(IJKM+1)+Y(IJKM-NJM)-
     *            Y(IJMK)-Y(IMJK-1)-Y(IJKM-1)-Y(IJKM-NJ-1))
      DYZD=HAF*(Y(IJK)-Y(IJKM)+Y(IMJK)-Y(IJKM-NJ))
      DZKS=HAF*(Z(IJK)-Z(IMJK)+Z(IJKM)-Z(IJKM-NJ))
      DZET=AHT*(Z(IJPK)+Z(IMJK+1)+Z(IJKM+1)+Z(IJKM-NJM)-
     *            Z(IJMK)-Z(IMJK-1)-Z(IJKM-1)-Z(IJKM-NJ-1))
      DZZD=HAF*(Z(IJK)-Z(IJKM)+Z(IMJK)-Z(IJKM-NJ))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
      ARN=B21**2+B22**2+B23**2
      DS=DN
      DN=GAMN*ARN
      ANN=AMIN1(F2(IJK),0.)*FY(IJPK)*G
      ASS=-AMAX1(F2(IJMK),0.)*(1.-FYSS)*G
      AN1=-AMIN1(F2(IJMK),0.)*FYN*G
      AN(IJK)=DN-AMIN1(F2(IJK),0.)-ANN
      AS1=AMAX1(F2(IJK),0.)*(1.-FY(IJMK))*G
      AS(IJK)=DS+AMAX1(F2(IJMK),0.)-ASS
C
      PHIN=PHI(IJK)*FYS+PHI(IJPK)*FYN
      PHITN=(PHI(IJKP)*FYS+PHI(IJKP+1)*FYN)*FZT+PHIN*FZB
      PHIBN=(PHI(IJKM)*FYS+PHI(IJKM+1)*FYN)*(1.-FZ(IJKM))+PHIN*FZ(IJKM)
      PHIEN=(PHI(IPJK)*FYS+PHI(IPJK+1)*FYN)*FXE+PHIN*FXW
      PHIWN=(PHI(IMJK)*FYS+PHI(IMJK+1)*FYN)*(1.-FX(IMJK))+PHIN*FX(IMJK)
      SUS=SUN
      SUN=GAMN*((B11*B21+B12*B22+B13*B23)*(PHIEN-PHIWN)+
     *          (B31*B21+B32*B22+B33*B23)*(PHITN-PHIBN))
C
C-----TOP  CELL - FACE
C
```

```
      DXKS=HAF*(X(IJK)-X(IMJK)+X(IJMK)-X(IMJK-1))
      DXET=HAF*(X(IJK)-X(IJMK)+X(IMJK)-X(IMJK-1))
      DXZD=AHT*(X(IJKP)+X(IJKP-1)+X(IJKP-NJ)+X(IJKP-NJ-1)-
     *          X(IJKM)-X(IJKM-1)-X(IJKM-NJ)-X(IJKM-NJ-1))
      DYKS=HAF*(Y(IJK)-Y(IMJK)+Y(IJMK)-Y(IMJK-1))
      DYET=HAF*(Y(IJK)-Y(IJMK)+Y(IMJK)-Y(IMJK-1))
      DYZD=AHT*(Y(IJKP)+Y(IJKP-1)+Y(IJKP-NJ)+Y(IJKP-NJ-1)-
     *          Y(IJKM)-Y(IJKM-1)-Y(IJKM-NJ)-Y(IJKM-NJ-1))
      DZKS=HAF*(Z(IJK)-Z(IMJK)+Z(IJMK)-Z(IMJK-1))
      DZET=HAF*(Z(IJK)-Z(IJMK)+Z(IMJK)-Z(IMJK-1))
      DZZD=AHT*(Z(IJKP)+Z(IJKP-1)+Z(IJKP-NJ)+Z(IJKP-NJ-1)-
     *          Z(IJKM)-Z(IJKM-1)-Z(IJKM-NJ)-Z(IJKM-NJ-1))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
      ART=B31**2+B32**2+B33**2
      DT=GAMT*ART
      ATT=AMIN1(F3(IJK),0.)*FZ(IJKP)*G
      ABB=-AMAX1(F3(IJKM),0.)*(1.-FZBB(IJ))*G
      AT1=-AMIN1(F3(IJKM),0.)*FZT*G
      AT(IJK)=DT-AMIN1(F3(IJK),0.)-ATT
      AB1=AMAX1(F3(IJK),0.)*(1.-FZ(IJKM))*G
      AB(IJK)=DB(IJ)+AMAX1(F3(IJKM),0.)-ABB
C
      PHIT=PHI(IJK)*FZB+PHI(IJKP)*FZT
      PHIET=(PHI(IPJK)*FZB+PHI(IJKP+NJ)*FZT)*FXE+PHIT*FXW
      PHIWT=(PHI(IMJK)*FZB+PHI(IJKP-NJ)*FZT)*(1.-FX(IMJK))+PHIT*FX(IMJK)
      PHINT=(PHI(IJPK)*FZB+PHI(IJKP+1)*FZT)*FYN+PHIT*FYS
      PHIST=(PHI(IJMK)*FZB+PHI(IJKP-1)*FZT)*(1.-FY(IJMK))+PHIT*FY(IJMK)
      SUT=GAMT*((B11*B31+B12*B32+B13*B33)*(PHIET-PHIWT)+
     *          (B21*B31+B22*B32+B23*B33)*(PHINT-PHIST))
      BT(IJK)=SUE-SUW(J)+SUN-SUS+SUT-SUB(IJ)
      BT(IJK) = AMAX1(0.0,BT(IJK))
      BT(IJK)=BT(IJK)+AEE*PHI(IPJK+NJ)+AWW*PHI(IMJK-NJ)+ANN*PHI(IJPK+1)+
     *           ASS*PHI(IJMK-1)+ATT*PHI(IJKP+NIJ)+ABB*PHI(IJKM-NIJ)+
     *           AE1*PHI(IPJK)+AW1*PHI(IMJK)+AN1*PHI(IJPK)+AS1*PHI(IJMK)+
     *           AT1*PHI(IJKP)+AB1*PHI(IJKM)
      SP(IJK)=AEE+AWW+ANN+ASS+ATT+ABB+AE1+AW1+AN1+AS1+AT1+AB1
      SUB(IJ)=SUT
      SUW(J)=SUE
      FZBB(IJ)=FZ(IJKM)
      FXWW(J)=FX(IMJK)
      FYSS=FY(IJMK)
      DW(J)=DE
      DB(IJ)=DT
C-----GENERATION OF TURB. KINETIC ENERGY
      DXKS=QTR*(X(IJK)+X(IJMK)+X(IJKM)+X(IJKM-1)-X(IMJK)-X(IMJK-1)-
     *          X(IJKM-NJ)-X(IJKM-NJ-1))
      DXET=QTR*(X(IJK)+X(IJKM)+X(IMJK)+X(IJKM-NJ)-
     *          X(IJMK)-X(IMJK-1)-X(IJKM-1)-X(IJKM-NJ-1))
      DXZD=QTR*(X(IJK)+X(IJMK)+X(IMJK)+X(IMJK-1)-X(IJKM)-
     *          X(IJKM-NJ)-X(IJKM-1)-X(IJKM-NJ-1))
      DYKS=QTR*(Y(IJK)+Y(IJMK)+Y(IJKM)+Y(IJKM-1)-Y(IMJK)-Y(IMJK-1)-
     *          Y(IJKM-NJ)-Y(IJKM-NJ-1))
      DYET=QTR*(Y(IJK)+Y(IJKM)+Y(IMJK)+Y(IJKM-NJ)-
     *          Y(IJMK)-Y(IMJK-1)-Y(IJKM-1)-Y(IJKM-NJ-1))
      DYZD=QTR*(Y(IJK)+Y(IJMK)+Y(IMJK)+Y(IMJK-1)-Y(IJKM)-
     *          Y(IJKM-NJ)-Y(IJKM-1)-Y(IJKM-NJ-1))
      DZKS=QTR*(Z(IJK)+Z(IJMK)+Z(IJKM)+Z(IJKM-1)-Z(IMJK)-Z(IMJK-1)-
     *          Z(IJKM-NJ)-Z(IJKM-NJ-1))
      DZET=QTR*(Z(IJK)+Z(IJKM)+Z(IMJK)+Z(IJKM-NJ)-
     *          Z(IJMK)-Z(IMJK-1)-Z(IJKM-1)-Z(IJKM-NJ-1))
      DZZD=QTR*(Z(IJK)+Z(IJMK)+Z(IMJK)+Z(IMJK-1)-Z(IJKM)-
```



```
     *          Z(IJKM-NJ)-Z(IJKM-1)-Z(IJKM-NJ-1))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
C
      UEW=U(IPJK)*FXE-U(IMJK)*(1.-FX(IMJK))+U(IJK)*(FXW-FX(IMJK))
      UNS=U(IJPK)*FYN-U(IJMK)*(1.-FY(IJMK))+U(IJK)*(FYS-FY(IJMK))
      UTB=U(IJKP)*FZT-U(IJKM)*(1.-FZ(IJKM))+U(IJK)*(FZB-FZ(IJKM))
      VEW=V(IPJK)*FXE-V(IMJK)*(1.-FX(IMJK))+V(IJK)*(FXW-FX(IMJK))
      VNS=V(IJPK)*FYN-V(IJMK)*(1.-FY(IJMK))+V(IJK)*(FYS-FY(IJMK))
      VTB=V(IJKP)*FZT-V(IJKM)*(1.-FZ(IJKM))+V(IJK)*(FZB-FZ(IJKM))
      WEW=W(IPJK)*FXE-W(IMJK)*(1.-FX(IMJK))+W(IJK)*(FXW-FX(IMJK))
      WNS=W(IJPK)*FYN-W(IJMK)*(1.-FY(IJMK))+W(IJK)*(FYS-FY(IJMK))
      WTB=W(IJKP)*FZT-W(IJKM)*(1.-FZ(IJKM))+W(IJK)*(FZB-FZ(IJKM))
C
      GEN(IJK)=(2.*((B11*UEW+B21*UNS+B31*UTB)**2+
     *      (B12*VEW+B22*VNS+B32*VTB)**2+(B13*WEW+B23*WNS+B33*WTB)**2
     *   )+(B12*UEW+B22*UNS+B32*UTB+B11*VEW+B21*VNS+B31*VTB)**2+
     *      (B13*UEW+B23*UNS+B33*UTB+B11*WEW+B21*WNS+B31*WTB)**2+
     *      (B13*VEW+B23*VNS+B33*VTB+B12*WEW+B22*WNS+B32*WTB)**2)
     *      /(VOL(IJK)**2)
C
      IF(IPHI.EQ.2) GO TO 102
C
C-----TURBULENCE KINETIC ENERGY - SOURCE TERMS
C
      SU(IJK)=BT(IJK)+GEN(IJK)*VOL(IJK)*(VIS(IJK)-VISCOS)
      SP(IJK)=SP(IJK)+VOL(IJK)*CMU*DENSIT*DENSIT*TE(IJK)/
     *             (VIS(IJK)-VISCOS+SMALL)
C
      GO TO 100
C
C-----DISSIPATION OF TURB. KIN. ENERGY SOURCE TERMS
C
  102 CONTINUE
      SU(IJK)=BT(IJK)+C1*GEN(IJK)*VOL(IJK)*DENSIT*CMU*TE(IJK)
      SP(IJK)=SP(IJK)+C2*VOL(IJK)*DENSIT*DENSIT*CMU*TE(IJK)/
     *             (VIS(IJK)-VISCOS+SMALL)
C
  100 CONTINUE
C
C-----PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS
C
      IF(IPHI.EQ.1) CALL MODKE(BT)
      IF(IPHI.EQ.2) CALL MODED
C
      DO 200 K=2,NKM
      LKK=LK(K)
      DO 200 I=2,NIM
      LIK=LKK+LI(I)
      DO 200 J=2,NJM
      IJK=LIK+J
      AP(IJK)=AE(IJK)+AW(IJK)+AN(IJK)+AS(IJK)+AT(IJK)+AB(IJK)+SP(IJK)
      AP(IJK)=AP(IJK)*URFPHI
      SU(IJK)=SU(IJK)+(1.-URFPHI)*AP(IJK)*PHI(IJK)
  200 CONTINUE
C
C-----SOLVING F.D. EQUATIONS
C
      CALL SOLSIP(PHI,IPHI)
C
      RETURN
      END
```

```
C
C**********************************************************************
      SUBROUTINE SOLSIP(PHI,IPHI)
C**********************************************************************
      INCLUDE 'kemod.h'
C
      DIMENSION PHI(1),BB(JXYZ),BS(JXYZ),BW(JXYZ),BP(JXYZ),API(JXYZ)
      LOGICAL ITEST
C
C-----CALCULATE COEFFICIENTS OF  L  AND  U  MATRICES
C
       DO IJK=1,NIJK
        API(IJK) = 1.0/(AP(IJK)+SMALL)
        AP(IJK) = 1.0
        AS(IJK) = AS(IJK)*API(IJK)
        AN(IJK) = AN(IJK)*API(IJK)
        AW(IJK) = AW(IJK)*API(IJK)
        AE(IJK) = AE(IJK)*API(IJK)
        AB(IJK) = AB(IJK)*API(IJK)
        AT(IJK) = AT(IJK)*API(IJK)
        SU(IJK) = SU(IJK)*API(IJK)
       ENDDO
       ENDIF
C
      L=0
      DO 20 K=2, NKM
      LKK=LK(K)
      DO 20 I=2, NIM
      LKI=LI(I)+LKK
      DO 20 J=2, NJM
      IJK=LKI+J
      IMJK=IJK-NJ
      IJKM=IJK-NIJ
      IJMK=IJK-1
      BB(IJK)=-AB(IJK)/(1.+ALFA*(BN(IJKM)+BE(IJKM)))
      BW(IJK)=-AW(IJK)/(1.+ALFA*(BN(IMJK)+BT(IMJK)))
      BS(IJK)=-AS(IJK)/(1.+ALFA*(BE(IJMK)+BT(IJMK)))
      POM1=ALFA*(BB(IJK)*BN(IJKM)+BW(IJK)*BN(IMJK))
      POM2=ALFA*(BB(IJK)*BE(IJKM)+BS(IJK)*BE(IJMK))
      POM3=ALFA*(BW(IJK)*BT(IMJK)+BS(IJK)*BT(IJMK))
      BP(IJK)=AP(IJK)+POM1+POM2+POM3-BB(IJK)*BT(IJKM)-BW(IJK)*BE(IMJK)
     *       -BS(IJK)*BN(IJMK)
      BPR=1./(BP(IJK)+SMALL)
      BN(IJK)=(-AN(IJK)-POM1)*BPR
      BE(IJK)=(-AE(IJK)-POM2)*BPR
      BT(IJK)=(-AT(IJK)-POM3)*BPR
   20 CONTINUE
C-----CALCULATE RESIDUALS
   25 L=L+1
      RES1=0.0
      DO 30 K=2,NKM
      LKK=LK(K)
      DO 30 I=2,NIM
      LKI=LKK+LI(I)
      DO 30 J=2,NJM
      IJK=LKI+J
      RES(IJK)=AE(IJK)*PHI(IJK+NJ)+AW(IJK)*PHI(IJK-NJ)+AN(IJK)*
     *         PHI(IJK+1)+AS(IJK)*PHI(IJK-1)+AT(IJK)*PHI(IJK+NIJ)+
     *         AB(IJK)*PHI(IJK-NIJ)+SU(IJK)-AP(IJK)*PHI(IJK)
      RES1=RES1+ABS(RES(IJK))
      RES(IJK)=(RES(IJK)-BB(IJK)*RES(IJK-NIJ)-BW(IJK)*RES(IJK-NJ)-
     *          BS(IJK)*RES(IJK-1))/(BP(IJK)+SMALL)
   30 CONTINUE
      IF(L.EQ.1) RESOR(IPHI)=RES1
      RSM=RES1/RESOR(IPHI)
C-----UPDATE VARIABLES
      DO 40 KK=2, NKM
      K=NKM+2-KK
      LKK=LK(K)
      DO 40 II=2,NIM
      I=NIM+2-II
```



```
      LIK=LKK+LI(I)
      DO 40 JJ=2,NJM
      J=NJM+2-JJ
      IJK=LIK+J
      RES(IJK)=RES(IJK)-BN(IJK)*RES(IJK+1)-BE(IJK)*RES(IJK+NJ)-
     *            BT(IJK)*RES(IJK+NIJ)
      PHI(IJK)=PHI(IJK)+RES(IJK)
   40 CONTINUE
      IF(ITEST) PRINT 60, L,RES1
      IF(RSM.GT.SOR(IPHI).AND.L.LT.NSWP(IPHI)) GO TO 25
      IF(IPHI.EQ.IP.AND.L.EQ.NSWP(IP)) WRITE(6,61)
   60 FORMAT(5X,I3,' SWEEP, RES = ',E14.6)
   61 FORMAT(10X,'PRESSURE CORR. NOT CONVERGED')
      RETURN
      END
C
C-----------------------
      SUBROUTINE MODVIS
C-----------------------
      INCLUDE 'kemod.h'
      DATA URFVIS/0.5/
      DO 100 K=2,NKM
      LKK=LK(K)
      DO 100 I=2,NIM
      LIK=LKK+LI(I)
      DO 100 J=2,NJM
      IJK=LIK+J
      VISOLD=VIS(IJK)
      VIS(IJK)=VISCOS
      IF(ED(IJK).GT.SMALL)
     & VIS(IJK)=DENSIT*TE(IJK)**2*CMU/ED(IJK)+VISCOS
      VIS(IJK)=URFVIS*VIS(IJK)+(1.-URFVIS)*VISOLD
  100 CONTINUE
      RETURN
      END
C
C------------------------
      SUBROUTINE MODK (BT)
C------------------------
      INCLUDE 'kemod.h'
      DATA CAPPA/0.4197/
      DIMENSION BT(JXYZ)
      INTEGER BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
C
C-----TURB. KINETIC ENERGY BOUNDARY CONDITIONS
C-----BOTTOM AND TOP BOUNDARIES
      DO 600 I=2,NIM
      LII = LI(I)

      DO 610 J=2,NJM
      IJ = LII+J

      IJK = LK(2) + IJ

      GO TO (10,20,30,40) BCFB(IJ)
C  INLET
10    SU(IJK) = SU(IJK) + AB(IJK)*TE(IJK-NIJ)
      SP(IJK) = SP(IJK) + AB(IJK)
      GO TO 20
C SYMMETRY
30    CONTINUE
      GO TO 20
C WALL
40    SU(IJK) = BT(IJK) + GENTB(IJ)*VOL(IJK)

20    AB(IJK)=0.
C
      IJK=LK(NKM)+IJ
      GO TO (11,21,31,41) BCFT(IJ)
C  INLET
11    SU(IJK) = SU(IJK) + AT(IJK)*TE(IJK+NIJ)
```

```
      SP(IJK) = SP(IJK) + AT(IJK)
      GO TO 21
C  SYMMETRY
31    CONTINUE
      GO TO 21
C  WALL
41    SU(IJK) = BT(IJK) + GENTT(IJ)*VOL(IJK)

21    AT(IJK)=0.
  610 CONTINUE
C
C-----SOUTH AND NORTH BOUNDARIES
C
      DO 620 K=2,NKM
      IK=(I-1)*NK+K
      IJK=LK(K)+LII+2
      GO TO (12,22,32,42) BCFS(IK)
C  INLET
12    SU(IJK) = SU(IJK) + AS(IJK)*TE(IJK-1)
      SP(IJK) = SP(IJK) + AS(IJK)
      GO TO 22
C  SYMMETRY
32    CONTINUE
      GO TO 22
C  WALL
42    SU(IJK) = BT(IJK) + GENTS(IK)*VOL(IJK)

22    AS(IJK) = 0.

C
      IJK=LK(K)+LII+NJM
      GO TO (13,23,33,43) BCFN(IK)
C  INLET
13    SU(IJK) = SU(IJK) + AN(IJK)*TE(IJK+1)
      SP(IJK) = SP(IJK) + AN(IJK)
      GO TO 23
C  SYMMETRY
33    CONTINUE
      GO TO 23
C  WALL
43    SU(IJK) = BT(IJK) + GENTN(IK)*VOL(IJK)

23    AN(IJK)=0.
  620 CONTINUE
  600 CONTINUE
C
C-----WEST AND EAST BOUNDARIES
C
      DO 630 K=2,NKM
      LKK = LK(K)

      KK = (K-1)*NJ
      DO 630 J=2,NJM
      IJK = LKK + LI(2) + J

      JK = KK + J
      GO TO (14,24,34,44) BCFW(JK)

C  INLET
14    SP(IJK) = SP(IJK) + AW(IJK)

      SU(IJK) = SU(IJK) + AW(IJK)*TE(IJK-NJ)
      GO TO 24
C  SYMMETRY
34    CONTINUE
      GO TO 24
C  WALL
44    SU(IJK) = BT(IJK) + GENTW(JK)*VOL(IJK)
24    AW(IJK)=0.
C
      IJK = LKK + LI(NIM) + J
```



```
      GO TO (15,25,35,45) BCFE(JK)
C  INLET
15    SP(IJK) = SP(IJK) + AE(IJK)

      SU(IJK) = SU(IJK) + AE(IJK)*TE(IJK+NJ)
      GO TO 25
C  SYMMETRY
35    CONTINUE
      GO TO 25
C  WALL
45    SU(IJK) = BT(IJK) + GENTE(JK)*VOL(IJK)
25    AE(IJK)=0.
C
  630 CONTINUE
      RETURN
      END
C
C---------------------
      SUBROUTINE MODED
C---------------------
      INCLUDE 'kemod.h'
C
C-----DISSIPATION OF TURB. EN. BOUNDARY CONDITIONS
      DATA CAPPA/0.4197/
      CMu25=SQRT(SQRT(CMU))
      CMU75=CMu25**3

      CONST=GREAT*CM75/CAPPA
C
C-----BOTTOM AND TOP BOUNDARIES
C
      DO 600 I=2,NIM
      LII = LI(I)

      DO 610 J=2,NJM
      IJ = LII+J

      IJK = LK(2) + IJ

      GO TO (10,20,30,40) BCFB(IJ)
C  INLET
10    SU(IJK) = SU(IJK) + AB(IJK)*ED(IJK-NIJ)
      SP(IJK) = SP(IJK) + AB(IJK)
      GO TO 20
C SYMMETRY
30    CONTINUE
      GO TO 20
C WALL
40    TE(IJK) = AMAX1(TE(IJK),-TE(IJK))

      SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNB(IJ)

      SP(IJK) = GREAT
20    AB(IJK)=0.
C
      IJK=LK(NKM)+IJ
      GO TO (11,21,31,41) BCFT(IJ)
C  INLET
11    SU(IJK) = SU(IJK) + AT(IJK)*ED(IJK+NIJ)
      SP(IJK) = SP(IJK) + AT(IJK)
      GO TO 21
C  SYMMETRY
31    CONTINUE
      GO TO 21
C  WALL
41    TE(IJK) = AMAX1(TE(IJK),-TE(IJK))

      SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNT(IJ)

      SP(IJK) = GREAT
```

```
21    AT(IJK)=0.
  610 CONTINUE
C
C-----SOUTH AND NORTH BOUNDARIES
C
      DO 620 K=2,NKM
      IK=(I-1)*NK+K
      IJK=LK(K)+LII+2
      GO TO (12,22,32,42) BCFS(IK)
C  INLET
12    SU(IJK) = SU(IJK) + AS(IJK)*ED(IJK-1)
      SP(IJK) = SP(IJK) + AS(IJK)
      GO TO 22
C  SYMMETRY
32    CONTINUE
      GO TO 22
C  WALL
42    TE(IJK) = AMAX1(TE(IJK),-TE(IJK))

      SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNS(IK)

      SP(IJK) = GREAT
22    AS(IJK) = 0.

C
      IJK=LK(K)+LII+NJM
      GO TO (13,23,33,43) BCFN(IK)
C  INLET
13    SU(IJK) = SU(IJK) + AN(IJK)*ED(IJK+1)
      SP(IJK) = SP(IJK) + AN(IJK)
      GO TO 23
C  SYMMETRY
33    CONTINUE
      GO TO 23
C  WALL
43    TE(IJK) = AMAX1(TE(IJK),-TE(IJK))

      SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNN(IK)

      SP(IJK) = GREAT
23    AN(IJK)=0.
  620 CONTINUE
  600 CONTINUE
C
C-----WEST AND EAST BOUNDARIES
C
      DO 630 K=2,NKM
      LKK = LK(K)

      KK = (K-1)*NJ
      DO 630 J=2,NJM
      IJK = LKK + LI(2) + J

      JK = KK + J
      GO TO (14,24,34,44) BCFW(JK)
C  INLET
14    SP(IJK) = SP(IJK) + AW(IJK)

      SU(IJK) = SU(IJK) + AW(IJK)*ED(IJK-NJ)
      GO TO 24
C  SYMMETRY
34    CONTINUE
      GO TO 24
C  WALL
44    TE(IJK) = AMAX1(TE(IJK),-TE(IJK))

      SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNW(JK)

      SP(IJK) = GREAT
24    AW(IJK)=0.
C
```



```
      IJK = LKK + LI(NIM) + J

      GO TO (15,25,35,45) BCFE(JK)
C  INLET
15    SP(IJK) = SP(IJK) + AE(IJK)

      SU(IJK) = SU(IJK) + AE(IJK)*ED(IJK+NJ)
      GO TO 25
C  SYMMETRY
35    CONTINUE
      GO TO 25
C  WALL
45    TE(IJK) = AMAX1(TE(IJK),-TE(IJK))

      SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNE(JK)

      SP(IJK) = GREAT
25    AE(IJK)=0.
C
  630 CONTINUE
      RETURN
      END
C
C------------------------------
      SUBROUTINE MODIFY (SU,BP)
C------------------------------
      INCLUDE 'kemod.h'
      DIMENSION SU(JXYZ),BP(JXYZ)
C
      DATA CMU25,CAPPA,ELOG/0.5477,0.4197,9.0/
C
C-----BOTTOM BOUNDARY ( WALL )
      DO I=2,NIM
      LII=LI(I)
      DO J=2,NJM
      IJ=LII+J
      IJK=LK(2)+IJ
      IJKM=IJK-NIJ
      IMJKM=IJKM-NJ
      IF(BCFB(IJ).EQ.4) THEN
      DX1=X(IJKM-1)-X(IMJKM)
      DY1=Y(IJKM-1)-Y(IMJKM)
      DZ1=Z(IJKM-1)-Z(IMJKM)
      DX2=X(IJKM)-X(IMJKM-1)
      DY2=Y(IJKM)-Y(IMJKM-1)
      DZ2=Z(IJKM)-Z(IMJKM-1)
      LW=IJK
      LB=IJK-NIJ
      DELN=DNB(IJ)
      CALL WALLFN (LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPA,DEN,GENTE,DELN,
     &  CMU25,VISC,ELOG,YPLS,SPU,SUU,SPV,SUV,SPW,SUWW,TAU)
      BP(IJK)=BP(IJK)+SPU
      SU(IJK)=SU(IJK)+SUU
      SPVB(IJ)=SPV
      SPWB(IJ)=SPW
      SUVB(IJ)=SUV
      SUWB(IJ)=SUWW
      GENTB(IJ)=GENTE
      AB(IJK)=0.
      ENDIF
      ENDDO
      ENDDO
C-----TOP BOUNDARY
      DO I=2,NIM
      LII = LI(I)
      DO J=2,NJM
      IJ=LII+J
      IJK=LK(NKM)+IJ
      IMJK=IJK-NJ
      IF(BCFT(IJ).EQ.4) THEN
      DX1=X(IJK)-X(IMJK-1)
```

```
      DY1=Y(IJK)-Y(IMJK-1)
      DZ1=Z(IJK)-Z(IMJK-1)
      DX2=X(IJK-1)-X(IMJK)
      DY2=Y(IJK-1)-Y(IMJK)
      DZ2=Z(IJK-1)-Z(IMJK)
      LW=IJK
      LB=IJK+NIJ
      DELN=DNT(IJ)
      CALL WALLFN (LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPA,DEN,GENTE,DELN,
     &  CMU25,VISC,ELOG,YPLS,SPU,SUU,SPV,SUV,SPW,SUWW,TAU)
      BP(IJK)=BP(IJK)+SPU
      SU(IJK)=SU(IJK)+SUU
      SPVT(IJ)=SPV
      SPWT(IJ)=SPW
      SUVT(IJ)=SUV
      SUWT(IJ)=SUWW
      GENTT(IJ)=GENTE
      AT(IJK)=0.0
      ENDIF
      ENDDO
      ENDDO
C-----SOUTH BOUNDARY
      DO I=2,NIM
       II = (I-1)*NK
       DO K=2,NKM
        IK = II + K
        IJK = LK(K) + LI(I) + 2
        IJKM = IJK - NIJ
        IMJK = IJK - NJ
        IMJKM = IJKM - NJ
        IJMK = IJK - 1
      IF(BCFS(IK).EQ.4) THEN
        DX1 = X(IMJK-1) - X(IJKM-1)
        DY1 = Y(IMJK-1) - Y(IJKM-1)
        DZ1 = Z(IMJK-1) - Z(IJKM-1)
        DX2 = X(IJK-1) - X(IMJKM-1)
        DY2 = Y(IJK-1) - Y(IMJKM-1)
        DZ2 = Z(IJK-1) - Z(IMJKM-1)
        LW = IJK
        LB = IJK - 1
        DELN = DNS(IK)
        CALL WALLFN (LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPA,DEN,GENTE,DELN,
     &  CMU25,VISC,ELOG,YPLS,SPU,SUU,SPV,SUV,SPW,SUWW,TAU)
        BP(IJK) = BP(IJK) + SPU
        SU(IJK) = SU(IJK) + SUU
        SPVS(IK) = SPV
        SPWS(IK) = SPW
        SUVS(IK) = SUV
        SUWS(IK) = SUW
        GENTS(IK) = GENTE
        AS(IJK)=0.0
      ENDIF
      ENDDO
      ENDDO
C-----NORTH BOUNDARY
      DO I=2,NIM
       DO K=2,NKM
        IK = (I-1)*NK + K
        IJK = LK(K) + LI(I) + NJM
        IJKM = IJK - NIJ
        IMJK = IJK - NJ
        IMJKM = IJKM - NJ
        IJPK = IJK + 1
       IF(BCFN(IK).EQ.4) THEN
        DX1 = X(IJK) - X(IMJKM)
        DY1 = Y(IJK) - Y(IMJKM)
        DZ1 = Z(IJK) - Z(IMJKM)
        DX2 = X(IMJK) - X(IJKM)
        DY2 = Y(IMJK) - Y(IJKM)
        DZ2 = Z(IMJK) - Z(IJKM)
        LW = IJK
```



```
        LB = IJK + 1
        DELN = DNN(IK)
        CALL WALLFN (LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPA,DEN,GENTE,DELN,
     &  CMU25,VISC,ELOG,YPLS,SPU,SUU,SPV,SUV,SPW,SUWW,TAU)
        BP(IJK) = BP(IJK) + SPU
        SU(IJK) = SU(IJK) + SUU
        SPVN(IK) = SPV
        SPWN(IK) = SPW
        SUVN(IK) = SUV
        SUWN(IK) = SUW
        GENTN(IK) = GENT
        AN(IJK) = 0.0
       ENDIF
       ENDDO
      ENDDO
C-----WEST BOUNDARY
      DO K=2,NKM
       LKK = LK(K)
       KK = (K-1)*NJ
       DO J=2,NJM
        JK = KK + J
        IJK = LKK + LI(2) + J
        IMJK = IJK - NJ
        IJMK = IJK - 1
        IJKM = IJK - NIJ
        IJMKM = IJKM - 1
       IF(BCFW(JK).EQ.4) THEN
        DX1 = X(IJMK-NJ) - X(IJKM-NJ)
        DY1 = Y(IJMK-NJ) - Y(IJKM-NJ)
        DZ1 = Z(IJMK-NJ) - Z(IJKM-NJ)
        DX2 = X(IJK-NJ) - X(IJMKM-NJ)
        DY2 = Y(IJK-NJ) - Y(IJMKM-NJ)
        DZ2 = Z(IJK-NJ) - Z(IJMKM-NJ)
        LW = IJK
        LB = IJK - NJ
        DELN = DNW(JK)
        CALL WALLFN (LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPA,DEN,GENTE,DELN,
     &  CMU25,VISC,ELOG,YPLS,SPU,SUU,SPV,SUV,SPW,SUWW,TAU)
        BP(IJK) = BP(IJK) + SPU
        SU(IJK) = SU(IJK) + SUU
        SPVW(JK) = SPV
        SUVW(JK) = SUV
        SPWW(JK) = SPW
        SUWW(JK) = SUW
        GENTW(JK) = GENT
        AW(IJK)=0.0
       ENDIF
       ENDDO
      ENDDO
C-----EAST BOUNDARY
      DO K=2,NKM
       LKK = LK(K)
       KK = (K-1)*NJ
       DO J=2,NJM
        JK = KK + J
        IJK = LKK + LI(NIM) + J
        IJKM = IJK - NIJ
        IJMK = IJK - 1
        IJMKM = IJMK - NIJ
        IPJK = IJK + NJ
       IF(BCFE(JK).EQ.4) THEN
        DX1 = X(IJK) - X(IJMKM)
        DY1 = Y(IJK) - Y(IJMKM)
        DZ1 = Z(IJK) - Z(IJMKM)
        DX2 = X(IJMK) - X(IJKM)
        DY2 = Y(IJMK) - Y(IJKM)
        DZ2 = Z(IJMK) - Z(IJKM)
        LW = IJK
        LB = IJK + NJ
        DELN = DNE(JK)
        CALL WALLFN (LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPA,DEN,GENTE,DELN,
```

```
     &  CMU25,VISC,ELOG,YPLS,SPU,SUU,SPV,SUV,SPW,SUWW,TAU)
          BP(IJK) = BP(IJK) + SPU
          SU(IJK) = SU(IJK) + SUU
          SPVE(JK) = SPV
          SUVE(JK) = SUV
          SPWE(JK) = SPW
          SUWE(JK) = SUW
          GENTE(JK) = GENT
          AE(IJK)=0.0
         ENDIF
         ENDDO
        ENDDO
C
C--------------------------------------------------------------------------

        SUBROUTINE WALLFN(LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPA,
     &  DEN,GENTE,DELN,CMU25,VISC,ELOG,YPLS,SPU,SUU,
     &  SPV,SUV,SPW,SUWW,TAU)
C--------------------------------------------------------------------------

        INCLUDE 'kemod.h'
C-----WALL CELL FACE AREA
        XAN=DY1*DZ2-DY2*DZ1
        YAN=DX2*DZ1-DX1*DZ2
        ZAN=DX1*DY2-DX2*DY1
        ARN=0.5*SQRT(XAN**2+YAN**2+ZAN**2)
        ARNR=0.5/ARN
C-----COMPONENTS OF UNIT NORMAL VECTOR
        ALFAN=XAN*ARNR
        BETAN=YAN*ARNR
        GAMAN=ZAN*ARNR
C-----CALCULATE  Y+  AND  LAMBDA-WALL COEFF.
        CONST=DEN*CMU25*SQRT(TE(LW))
        YPLS=DELN*CONST/(VISC+1.E-30)
        TCOEF=VISC/DELN
        IF(YPLS.GT.11.63) TCOEF=CONST*CAPA/(ALOG(ELOG*YPLS))
        TAR=TCOEF*ARN
C-----SOURCE TERMS FOR VELOCITIES
        SPU=TAR*(1.-ALFAN**2)
        SPV=TAR*(1.-BETAN**2)
        SPW=TAR*(1.-GAMAN**2)
        SUU=TAR*ALFAN*(BETAN*V(LW)+GAMAN*W(LW))
        SUV=TAR*BETAN*(ALFAN*U(LW)+GAMAN*W(LW))
        SUWW=TAR*GAMAN*(ALFAN*U(LW)+BETAN*V(LW))
C-----VELOCITY PARALEL TO WALL
        UP=SQRT((SPU*U(LW)-SUU)**2+(SPV*V(LW)-SUV)**2+
     *           (SPW*W(LW)-SUWW)**2)/(TAR+1.E-30)
        UP=ABS(UP-SQRT(U(LB)**2+V(LB)**2+W(LB)**2))
C-----WALL SHEAR STRESS AND GENER. TERM
        SUU=SUU+TAR*U(LB)
        SUV=SUV+TAR*V(LB)
        SUWW=SUWW+TAR*W(LB)
        TAU=UP*TCOEF
        GENTE=TAU*UP/DELN
        RETURN
        END
C
C-------------------------------------------------------------------
C
C-- kemod.h
C
        PARAMETER (JX=70)
        PARAMETER (JY=41)
        PARAMETER (JZ=41)
        PARAMETER (JXY=JX*JY)
        PARAMETER (JXZ=JX*JZ)
        PARAMETER (JYZ=JY*JZ)
        PARAMETER (JXYZ=JX*JY*JZ)
C
        COMMON NI,NJ,NK,NIM,NJM,NKM,NIJ,NIK,NIJK,LK(JZ),LI(JX),IU,IV,IW,
     *         IP,ITE,IED,IVIS,IEN,NITER,MAXIT,ITSTEP,ITEST,ICAL(10),
```



```
     *         IMON,JMON,KMON,IJKMON,IJKPR,IDIR,NSWP(2),G,SOR(2)
     *        ,X(JXYZ),Y(JXYZ),Z(JXYZ),FX(JXYZ),FY(JXYZ),FZ(JXYZ),
     *         VOL(JXYZ),DP1(JXYZ),DP2(JXYZ),DP3(JXYZ)
     *        ,AE(JXYZ),AW(JXYZ),AN(JXYZ),AS(JXYZ),AT(JXYZ),AB(JXYZ),
     *         AP(JXYZ),SU(JXYZ),BE(JXYZ),BN(JXYZ),BT(JXYZ),RES(JXYZ),
     *         SWU(JXYZ),DB(JXY),DW(JY)
     *        ,RESOR(2),SNORIN(2),PRTINV(2),URF(2),HAF,QTR,AHT,OMEGA
     *        ,ALFA,DENSIT,SORMAX,VISCOS,READ1,WRIT7,IOBST,GREAT,SMALL,
     *         C1,C2,CAPPA,CMU,CMU25,CMU75,SUW(JY),SUB(JXY),UWB(JY),
     *         VWB(JY),WWB(JY),FZBB(JXY),FXWW(JY)
        COMMON /UVW/ P(JXYZ),PP(JXYZ),VIS(JXYZ),U(JXYZ),V(JXYZ),
     * W(JXYZ),TE(JXYZ),ED(JXYZ),T(1),F1(JXYZ),F2(JXYZ),F3(JXYZ)
```

CHAPTER 7

3D Algebraic Stress Turbulence Model

Table of Contents

7.1 Introduction

In this section a description is given of the three-dimensional Algebraic Stress turbulence Model (ASM) based on the work of Rodi [1]. The model is coded as a self contained computer program to compute turbulent flow quantities when interfaced with a CFD solver. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in the Appendix.

The module uses as input the mean flow properties, as computed by conventional CFD solvers, and calculates the Reynolds stresses, turbulent kinetic energy and the energy dissipation. It is structured to be self-contained and compatible with many CFD codes. The module has not been tested thoroughly due to the ending of the contract earlier than scheduled. Some testing of the module has been done at UAH but that also has been put on hold. However, the module as assembled is capable of interfacing with a number CFD solvers.

The module computes turbulent flow quantities in three-dimensional body-fitted geometry with or without rotation about any one of the three axis. The standard wall functions is used for the near wall treatment.

7.2 Theory and Model Equations

The Algebraic Stress (ASM) module discussed here is based on the work of Rodi [1]. The idea is to simplify or truncate the Reynolds stress equation by approximating the convective and diffusive transport of the Reynolds stresses $\overline{u_i u_j}$ in terms of the corresponding transport of turbulent energy. This allows the transport equation for the stresses to be expressed as a set of algebraic formulae containing the turbulence energy and its rate of dissipation as unknowns in the form:

$$\overline{u_i u_j} = \frac{k}{(P-\varepsilon)} \left[P_{ij} - \frac{2}{3} \delta_{ij} \varepsilon + \Phi_{ij} \right]$$

where P_{ij} = Production and $P = \frac{1}{2} P_{kk}$

Φ_{ij} = pressure-strain redistribution

$$\Phi_{ij} = \Phi_{ij,1} + \Phi_{ij,2} + \Phi_{ij,1w} + \Phi_{ij,2w}$$

Rotta's linear return-to-isotropy concept for the non-linear part

$$\Phi_{ij,1} = -C_1 \frac{\varepsilon}{k} (\overline{u_i u_j} - \frac{2}{3} k \, \delta_{ij})$$

is used and the "isotropization of production" concept for the linear "rapid" part

$$\Phi_{ij,2} = -C_2 \, (P_{ij} - \frac{2}{3} P \, \delta_{ij})$$

is used. Gibson and Launder [2] concept for the wall reflection terms is used as

$$\Phi_{ij,1w} = C_{1w} \, \rho \frac{\varepsilon}{k} (\overline{u_k u_m} \; n_k n_m \delta_{ij} - \frac{3}{2} \overline{u_k u_i} \; n_k n_j - \frac{3}{2} \overline{u_k u_j} \; n_k n_i) f$$

$$\Phi_{ij,2w} = C_{2w} \, (\Phi_{km,2} \; n_k n_m \delta_{ij} - \frac{3}{2} \Phi_{ik,2} \; n_k n_j - \frac{3}{2} \Phi_{jk,2} \; n_k n_i) f$$

where (n_i) is the wall-normal unit vector in the i-direction. The wall-distance function (f) represents the ratio of the turbulence length scale $(L_\varepsilon = \frac{k^{3/2}}{\varepsilon})$ and the wall distance and is given as

$$f = (\frac{C_m^{0.75} \; k^{1.5}}{K \; \varepsilon}) \frac{1}{\Delta n}$$

with Δn being the wall-normal distance.

The resulting set of algebraic equations for the Reynolds stresses can be arranged in the form

$$A_{ij} \; \overline{u^2} + B_{ij} \; \overline{v^2} + C_{ij} \, \overline{w^2} + D_{ij} \, \overline{u\,v} + E_{ij} \; \overline{vw} + F_{ij} \; \overline{uw} = G_{ij}$$

where A_{ij}, B_{ij}, C_{ij}, D_{ij}, E_{ij}, F_{ij}, and G_{ij} are functions of the mean and turbulent flow variables.

The above equation can be solved iteratively in the main flow solver. However, the algebraic system of equations is stiff and convergence difficulties are encountered when solved iteratively. Therefore, the set of equations was cast in the general matrix form $\underline{A} \; \underline{T} = \underline{B}$, where

$$
\underline{\boldsymbol{A}} = \begin{bmatrix}
\frac{3\varepsilon}{2\lambda k} + 2\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & -\frac{\partial W}{\partial z} & 2\frac{\partial U}{\partial y} - \frac{\partial V}{\partial x} - 6\,C_0\,\Omega_z & -(\frac{\partial V}{\partial z} + \frac{\partial W}{\partial y}) & 2\frac{\partial U}{\partial z} - \frac{\partial W}{\partial x} + 6\,C_0\,\Omega_y \\
-\frac{\partial U}{\partial x} & \frac{3\varepsilon}{2\lambda k} + 2\frac{\partial V}{\partial y} & -\frac{\partial W}{\partial z} & 2\frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} + 6\,C_0\,\Omega_z & 2\frac{\partial V}{\partial z} - \frac{\partial W}{\partial y} - 6\,C_0\,\Omega_x & -(\frac{\partial U}{\partial z} + \frac{\partial W}{\partial x}) \\
-\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & \frac{3\varepsilon}{2\lambda k} + 2\frac{\partial W}{\partial z} & -(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) & -\frac{\partial V}{\partial z} + 2\frac{\partial W}{\partial y} + 6\,C_0\,\Omega_x & -\frac{\partial U}{\partial z} + 2\frac{\partial W}{\partial x} - 6\,C_0\,\Omega_y \\
\frac{\partial V}{\partial x} + 2\,C_0\,\Omega_z & \frac{\partial U}{\partial y} - 2\,C_0\,\Omega_z & 0 & \frac{\varepsilon}{\lambda k} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} & \frac{\partial U}{\partial z} + 2\,C_0\,\Omega_y & \frac{\partial V}{\partial z} - 2\,C_0\,\Omega_x \\
0 & \frac{\partial W}{\partial y} + 2\,C_0\,\Omega_x & \frac{\partial V}{\partial z} - 2\,C_0\,\Omega_x & \frac{\partial W}{\partial x} - 2\,C_0\,\Omega_y & \frac{\varepsilon}{\lambda k} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} & \frac{\partial V}{\partial x} + 2\,C_0\,\Omega_z \\
\frac{\partial W}{\partial x} - 2\,C_0\,\Omega_y & 0 & \frac{\partial U}{\partial z} + 2\,C_0\,\Omega_y & \frac{\partial W}{\partial y} + 2\,C_0\,\Omega_x & \frac{\partial U}{\partial y} + 2\,C_0\,\Omega_z & \frac{\varepsilon}{\lambda k} + \frac{\partial U}{\partial x} + \frac{\partial W}{\partial z}
\end{bmatrix}
$$

$$
\underline{\boldsymbol{T}} = [\rho\,\overline{uu}\,,\ \rho\,\overline{vv}\,,\ \rho\,\overline{ww}\,,\ \rho\,\overline{uv}\,,\ \rho\,\overline{vw}\,,\ \rho\,\overline{uw}\ ]^T
$$

$$
\underline{\boldsymbol{B}} = \begin{matrix}
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}\,(\Phi_{11,1w} + \Phi_{11,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}\,(\Phi_{22,1w} + \Phi_{22,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}\,(\Phi_{33,1w} + \Phi_{33,2w}) \\
\frac{1}{(1-C_2)}\,(\Phi_{12,1w} + \Phi_{12,2w}) \\
\frac{1}{(1-C_2)}\,(\Phi_{23,1w} + \Phi_{23,2w}) \\
\frac{1}{(1-C_2)}\,(\Phi_{13,1w} + \Phi_{13,2w})
\end{matrix}
$$

where $\lambda = \dfrac{1-C_2}{C_1 - 1 + \dfrac{P}{\rho\varepsilon}}$

The matrix was inverted at each iteration step to obtain a converged solution.

REFERENCES

[1] Rodi, W., 'A New Algebraic Stress Relations for Calculating the Reynolds Stresses' Z. Ang. Math. und Mech., vol. 56, pp. 219-, 1976.

[4] Gibson M. and Launder B., 'Ground Effects on Pressure Fluctuations in the Atmosphere boundary Layer', J. Fluid Mech. vol. 86, pt. 3, pp. 491-., 1978.

APPENDIX F

3D Algebraic Stress Module Deck

ASMOD is a FORTRAN source code to solve 2D/Axisymmetric turbulent flow quantities using the algebraic stress model when interfaced with a main flow solver. The module consists of the main routine ASMOD that calls a number of subroutines to perform different functions that will be explained below.

Subroutine ASMMOD

This is basically the main routine that reads through its argument list different variables from the calling flow solver which are described below.

List of Argument Variable Names

INITASM	Initialization parameter that writes and sets variables
NIM	Number of grid nodes in the i (or x) direction
NJM	Number of grid nodes in the j (or y) direction
NKM	Number of grid nodes in the k (or z) direction
LI	LI(I)=(I-1)*NJ, dimensioned to NX. Calculated as in subroutine GRIDG of the 3D k-ε module.
LK	LK(K)=(K-1)*NI*NJ dimensioned to NZ. Calculated as in subroutine GRIDG of the 3D k-ε module.
FX	grid interpolation factor in the x-direction
FY	grid interpolation factor in the y-direction
FZ	grid interpolation factor in the z-direction
X	Grid node locations in the x or ξ-direction, dimensioned to X(JXYZ=NX*NY*NZ)
Y	Grid node locations in the y or η-direction, dimensioned to Y(JXYZ)
Z	Grid node locations in the z or ζ-direction, dimensioned to Z(JXYZ)
VOL	Control cell volume (similar to that calculated in GRIDG of k-ε module)
U	mean velocity in x or ξ-direction, dimensioned to U(JXYZ) (input from the flow solver)

V	Mean velocity in the y or η-direction, dimensioned yo V(JXYZ) (input from the flow solver)
W	Mean velocity in the z or ζ-direction, dimensioned yo W(JXYZ) (input from the flow solver)
VIS	Eddy viscosity
TE	Turbulent kinetic energy, dimensioned to TE(JXYZ) calculated in the module.
ED	Turbulent energy dissipation, dimensioned to ED(JXYZ)
U2	Normal Reynolds stress component $\overline{u^2}$, calculated in the module
V2	Normal Reynolds stress component $\overline{v^2}$, calculated in the module
W2	Normal Reynolds stress component $\overline{w^2}$, calculated in the module
UV	Shear stress component $\overline{uv}$, calculated in the module
VW	Shear stress component $\overline{vw}$,calculated in the module
UW	Shear stress component $\overline{uw}$, calculated in the module
GEN	Turbulent energy generation term
SUASM	Source term for the U-momentum equation due to Reynolds stress gradients. Calculated in the module and passed to the main solver.
SVASM	Source term for the V-momentum equation due to Reynolds stress gradients. Calculated in the module and passed to the main solver.
SWASM	Source term for the W-momentum equation due to Reynolds stress gradients. Calculated in the module and passed to the main solver.
BCFW	Boundary condition flag along the west boundary (or y-z plane). It must have one for each boundary node set to; 1-inlet, 2-outlet, 3-symmetry and 4-wall. For example for an outlet flow condition on the west boundary set BCFW to (NY*NZ)*2, and similarly for the other boundaries, dimensioned to BCFW(JYZ=NY*NZ) (input from flow solver)
BCFE	Boundary condition flag for the east boundary dimensioned to BCFE(JYZ)
BCFS	Boundary condition flag for the south boundary dimensioned to BCFS(JXZ)
BCFN	Boundary condition flag for the north boundary dimensioned to BCFN(JYZ)
BCFB	Boundary condition flag for the bottom boundary dimensioned to BCFB(JXY)
BCFT	Boundary condition flag for the top boundary dimensioned to BCFT(JXY)
GENTW	Turbulent generation terms calculated from the wall functions close to the wall in the west direction. Similarly for the other GENTE, GENTS....
OMX	Frame rotation term in the x-direction. Similarly OMY & OMZ in the y and z-directions respectively

DENSIT	Constant density
VISCOS	Kinematic viscosity

All dimensions considered are one-dimensional. The position of any node is defined as IJK = (I,J,K) = (I-1)*NJ + (K-1)*NJ + J, where NI, NJ and NK are the number of grid nodes in the X, Y and Z-directions respectively. It is assumed that grid related data such as control volumes and interpolation factors be passed to the module from an external grid generator, similar to the one listed in the 3D k-e module (Chapter 6).

Subroutine CALPIJ

This subroutine calculates the production terms of the individual stress components.

Subroutine CALUIUJ

This subroutine calculates the individual stress component from its algebraic equation. It sets the coefficients of the algebraic stress equations which are solved implicitly at each iteration step by inverting a 6x6 matrix.

Subroutine SORUVW

This subroutine calculates the source terms needed in the momentum equation of the main CFD solver due to Reynolds stress gradients.

Subroutine SOLV

This subroutine is a Gaussian elimination solver to invert a 6x6 matrices.

Subroutine WALSTRS

This subroutine calculates the Reynolds stresses near the walls based on wall functions.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     3D-ALGEBRAIC STRESS TURBULENCE MODULE                            C
C                                                                      C
C     Rocketdyne CFD Technology Center                                 C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE ASMMOD(INITASM,
     & NIM,NJM,NKM,LI,LK,  FX,FY,FZ,
     & X,Y,Z,  VOL,  U,V,W,  VIS,TE,ED,
     & U2,  V2,W2,  UV,  UW,  VW,  GEN,
     & SUASM,  SVASM,  SWASM,
     & BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,
     & GENTW,GENTE,GENTS,GENTN,GENTB,
     & OMX,OMY,OMZ,  DENSIT,VISCOS
     & )
C
C******************************************************************
C
C Driver module for ASM Turbulence Module
C
      INCLUDE    'param.h'
      INCLUDE    'asmmod.h'
C
      DIMENSION  LK(JZ),LI(JX),
     &           X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),
     &           FX(JXYZ),FY(JXYZ),FZ(JXYZ)
      DIMENSION  U(JXYZ),V(JXYZ),W(JXYZ),
     &           TE(JXYZ),ED(JXYZ),VIS(JXYZ)
      DIMENSION  U2(JXYZ),V2(JXYZ),W2(JXYZ),
     &           UV(JXYZ),UW(JXYZ),VW(JXYZ)
      DIMENSION  SUASM(JXYZ),SVASM(JXYZ),SWASM(JXYZ)
      DIMENSION  DUU1(JXYZ),DVV1(JXYZ),DWW1(JXYZ),DUV1(JXYZ),DVW1(JXYZ),
     &           DUW1(JXYZ),DUU2(JXYZ),DVV2(JXYZ),DWW2(JXYZ),DUV2(JXYZ),
     &           DVW2(JXYZ),DUW2(JXYZ),DUU3(JXYZ),DVV3(JXYZ),DWW3(JXYZ),
     &           DUV3(JXYZ),DVW3(JXYZ),DUW3(JXYZ)
      DIMENSION  DUUE(JXYZ),DUUN(JXYZ),  DUUT(JXYZ),
     &           DVVE(JXYZ),DVVN(JXYZ),  DVVT(JXYZ),
     &           DWWE(JXYZ),DWWN(JXYZ),  DWWT(JXYZ)
      DIMENSION  BCFW(JYZ),BCFE(JYZ),BCFS(JXZ),BCFN(JXZ),
     &           BCFB(JXY),BCFT(JXY)
      DIMENSION  GEN(JXYZ)
      DIMENSION  GENTB(JXY),GENTT(JXY),GENTS(JXZ),GENTN(JXZ),
     &           GENTW(JYZ),GENTE(JYZ)
C
      INTEGER    BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
      LOGICAL    IASM
C
      NI=NIM+1
      NJ=NJM+1
      NK=NKM+1
      NIJ=NI*NJ
      NJK=NJ*NK
      NIK=NI*NK
      NIJK=NI*NJ*NK
C
C
      IF(INITASM.EQ.1)THEN
C Read Inputs:
        WRITE(6,*)'READING INPUTS TO ASM TURBULENCE MODEL'
        OPEN(42,FILE='ASMINP',  STATUS='OLD')
        REWIND 42
        WRITE(6,*)'ENTER: C1ASM,  C2ASM,  C1P,C2P,  COMEGA'
        READ(42,*)C1ASM,C2ASM,C1P,C2P,COMEGA
        WRITE(6,*)'ENTER: WREFON,  RELT'
        READ(42,*)WREFON,  RELT
C INITIALIZE STUFF FOR MODULE
        DO 20  IJK=1,NIJK
          DUDX(IJK)=0.0
          DUDY(IJK)=0.0
```



```
          DUDZ(IJK)=0.0
          DVDX(IJK)=0.0
          DVDY(IJK)=0.0
          DVDZ(IJK)=0.0
          DWDX(IJK)=0.0
          DWDY(IJK)=0.0
          DWDZ(IJK)=0.0
          FUNX(IJK)=0.0
          FUNY(IJK)=0.0
          FUNZ(IJK)=0.0
          FUNXY(IJK)=0.0
          FUNXZ(IJK)=0.0
          FUNYZ(IJK)=0.0
 20     CONTINUE
      ENDIF
C
C
      CALL CALPIJ(NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &  X,Y,Z,VOL,  U,V,W,  U2,V2,W2,  UV,UW,VW,   GEN,
     &  GENTB,GENTT,GENTS,GENTN,GENTW,GENTE,
     &  BCFW,BCFE,BCFS,BCFN,BCFB,BCFT )
c
      CALL CALUIUJ(NIM,NJM,NKM,LK,LI,  U2,V2,W2,  UV,UW,VW,
     &  GENTB,GENTT,GENTS,GENTN,GENTW,GENTE,  GEN,
     &  BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,
     &  X,Y,Z,  Fx,FY,FZ,  U,V,W,  VOL,
     &  TE,ED,  VIS,  OMX,OMY,OMZ,  DENSIT,VISCOS)
C
      CALL SORUVW( NIM,NJM,NKM,LI,LK,  X,Y,Z,
     &  U2,V2,W2,  UV,UW,VW,
     &  SUASM,  SVASM,  SWASM,
     &  BCFW,  BCFE,  BCFS,BCFN,BCFB,BCFT)
C
      INITASM=-1
C
      RETURN
      END
C******************************************************************
C
      SUBROUTINE SORUVW( NIM,NJM,NKM,LI,LK,  X,Y,Z,
     &  U2,V2,W2,  UV,UW,VW,
     &  SUASM,  SVASM,  SWASM,
     &  DUU1,DVV1,DWW1,DUV1,DVW1,DUW1,DUU2,DVV2,DWW2,
     &  DUV2,DVW2,DUW2,DUU3,DVV3,DWW3,DUV3,DVW3,DUW3,
     &  DUUE,DUUN,DUUT,  DVVE,DVVN,DVVT,  DWWE,DWWN,DWWT,
     &  BCFW,  BCFE,  BCFS,BCFN,BCFB,BCFT)
C
C******************************************************************
C
      INCLUDE    'param.h'
      INCLUDE    'asmmod.h'
C
      DIMENSION  LK(JZ),LI(JX),
     &           X(JXYZ),Y(JXYZ),Z(JXYZ)
      DIMENSION  U2(JXYZ),V2(JXYZ),W2(JXYZ),
     &           UV(JXYZ),UW(JXYZ),VW(JXYZ)
      DIMENSION  BCFW(JYZ),BCFE(JYZ),BCFS(JXZ),BCFN(JXZ),
     &           BCFB(JXY),BCFT(JXY)
      DIMENSION  SUASM(JXYZ),SVASM(JXYZ),SWASM(JXYZ)
      DIMENSION  DUU1(JXYZ),DVV1(JXYZ),DWW1(JXYZ),DUV1(JXYZ),DVW1(JXYZ),
     &           DUW1(JXYZ),DUU2(JXYZ),DVV2(JXYZ),DWW2(JXYZ),DUV2(JXYZ),
     &           DVW2(JXYZ),DUW2(JXYZ),DUU3(JXYZ),DVV3(JXYZ),DWW3(JXYZ),
     &           DUV3(JXYZ),DVW3(JXYZ),DUW3(JXYZ)
      DIMENSION  DUUE(JXYZ),DUUN(JXYZ),  DUUT(JXYZ),
     &           DVVE(JXYZ),DVVN(JXYZ),  DVVT(JXYZ),
     &           DWWE(JXYZ),DWWN(JXYZ),  DWWT(JXYZ)
C
      INTEGER    BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
C
      DO 20 K=2,NKM
      DO 20 I=2,NIM
```

```
      LIK = LI(I) + LK(K)
      DO 20 J=2,NJM
C
      IJK  = LIK + J
      IPJK = IJK + NJ
      IMJK = IJK - NJ
      IJKP = IJK + NIJ
      IJKM = IJK - NIJ
      IJPK = IJK + 1
C
      DUUE(IJK)=U2(IPJK)-U2(IJK)
      DUUN(IJK)=U2(IJPK)-U2(IJK)
      DUUT(IJK)=U2(IJKP)-U2(IJK)
      DVVE(IJK)=V2(IPJK)-V2(IJK)
      DVVN(IJK)=V2(IJPK)-V2(IJK)
      DVVT(IJK)=V2(IJKP)-V2(IJK)
      DWWE(IJK)=W2(IPJK)-W2(IJK)
      DWWN(IJK)=W2(IJPK)-W2(IJK)
      DWWT(IJK)=W2(IJKP)-W2(IJK)
C
      DUU1(IJK) = 0.5*(U2(IPJK) - U2(IMJK))
      DVV1(IJK) = 0.5*(V2(IPJK) - V2(IMJK))
      DWW1(IJK) = 0.5*(W2(IPJK) - W2(IMJK))
      DUV1(IJK) = 0.5*(UV(IPJK) - UV(IMJK))
      DUW1(IJK) = 0.5*(UW(IPJK) - UW(IMJK))
      DVW1(IJK) = 0.5*(VW(IPJK) - VW(IMJK))
C
      DUU2(IJK) = 0.5*(U2(IJK+1)-U2(IJK-1))
      DVV2(IJK) = 0.5*(V2(IJK+1)-V2(IJK-1))
      DWW2(IJK) = 0.5*(W2(IJK+1)-W2(IJK-1))
      DUV2(IJK) = 0.5*(UV(IJK+1)-UV(IJK-1))
      DUW2(IJK) = 0.5*(UW(IJK+1)-UW(IJK-1))
      DVW2(IJK) = 0.5*(VW(IJK+1)-VW(IJK-1))
C
      DUU3(IJK) = 0.5*(U2(IJKP) - U2(IJKM))
      DVV3(IJK) = 0.5*(V2(IJKP) - V2(IJKM))
      DWW3(IJK) = 0.5*(W2(IJKP) - W2(IJKM))
      DUV3(IJK) = 0.5*(UV(IJKP) - UV(IJKM))
      DUW3(IJK) = 0.5*(UW(IJKP) - UW(IJKM))
      DVW3(IJK) = 0.5*(VW(IJKP) - VW(IJKM))
   20 CONTINUE
C
      DO 31 K=2,NKM
C ALONG WEST AND EAST BOUNDARY
      KK = (K-1)*NJ
      DO 32 J=2,NJM
      JK = KK + J
      IJK = LK(K) + LI(2) + J
      IPJK = IJK + NJ
      IPPJK = IPJK + NJ
      IF(BCFW(JK).NE.6) THEN
      DUU1(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IPJK) - U2(IPPJK))
      DVV1(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IPJK) - V2(IPPJK))
      DWW1(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IPJK) - W2(IPPJK))
      DUV1(IJK) = 0.5*(-3.0*UV(IJK) + 4.0*UV(IPJK) - UV(IPPJK))
      DUW1(IJK) = 0.5*(-3.0*UW(IJK) + 4.0*UW(IPJK) - UW(IPPJK))
      DVW1(IJK) = 0.5*(-3.0*VW(IJK) + 4.0*VW(IPJK) - VW(IPPJK))
      ENDIF
      IJK = LK(K) + LI(NIM) + J
      IMJK = IJK - NJ
      IMMJK = IMJK - NJ
      IF(BCFE(JK).NE.6) THEN
      DUU1(IJK) = 0.5*(U2(IMMJK) -4.0*U2(IMJK) + 3.0*U2(IJK))
      DVV1(IJK) = 0.5*(V2(IMMJK) -4.0*V2(IMJK) + 3.0*V2(IJK))
      DWW1(IJK) = 0.5*(W2(IMMJK) -4.0*W2(IMJK) + 3.0*W2(IJK))
      DUV1(IJK) = 0.5*(UV(IMMJK) -4.0*UV(IMJK) + 3.0*UV(IJK))
      DUW1(IJK) = 0.5*(UW(IMMJK) -4.0*UW(IMJK) + 3.0*UW(IJK))
      DVW1(IJK) = 0.5*(VW(IMMJK) -4.0*VW(IMJK) + 3.0*VW(IJK))
      ENDIF
   32 CONTINUE
C
```



```
C ALONG SOUTH AND NORTH BOUNDARY
C
      DO 33 I=2,NIM
      IK = (I-1)*NK + K
      IJK = LK(K) + LI(I) + 2
      IF(BCFS(IK).NE.6) THEN
      DUU2(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IJK+1) - U2(IJK+2) )
      DVV2(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IJK+1) - V2(IJK+2) )
      DWW2(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IJK+1) - W2(IJK+2) )
      DUV2(IJK) = 0.5*(-3.0*UV(IJK) + 4.0*UV(IJK+1) - UV(IJK+2) )
      DVW2(IJK) = 0.5*(-3.0*VW(IJK) + 4.0*VW(IJK+1) - VW(IJK+2) )
      DUW2(IJK) = 0.5*(-3.0*UW(IJK) + 4.0*UW(IJK+1) - UW(IJK+2) )
      ENDIF
      IJK = LK(K) + LI(I) + NJM
      IF(BCFN(IK).NE.6) THEN
      DUU2(IJK) = 0.5*( U2(IJK-2) - 4.0*U2(IJK-1) + 3.0*U2(IJK) )
      DVV2(IJK) = 0.5*( V2(IJK-2) - 4.0*V2(IJK-1) + 3.0*V2(IJK) )
      DWW2(IJK) = 0.5*( W2(IJK-2) - 4.0*W2(IJK-1) + 3.0*W2(IJK) )
      DUV2(IJK) = 0.5*( UV(IJK-2) - 4.0*UV(IJK-1) + 3.0*UV(IJK) )
      DVW2(IJK) = 0.5*( VW(IJK-2) - 4.0*VW(IJK-1) + 3.0*VW(IJK) )
      DUW2(IJK) = 0.5*( UW(IJK-2) - 4.0*UW(IJK-1) + 3.0*UW(IJK) )
      ENDIF
   33 CONTINUE
   31 CONTINUE
C
C  ALONG BOTTOM AND TOP BOUNDARY
C
      DO 34 I=2,NIM
      DO 34 J=2,NJM
      IJK = LI(I) + LK(2) + J
      IJKP = IJK + NIJ
      IJKPP = IJKP + NIJ
      IJ = LI(I) + J
      IF(BCFB(IJ).NE.6) THEN
      DUU3(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IJKP) - U2(IJKPP))
      DVV3(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IJKP) - V2(IJKPP))
      DWW3(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IJKP) - W2(IJKPP))
      DUV3(IJK) = 0.5*(-3.0*UV(IJK) + 4.0*UV(IJKP) - UV(IJKPP))
      DVW3(IJK) = 0.5*(-3.0*VW(IJK) + 4.0*VW(IJKP) - VW(IJKPP))
      DUW3(IJK) = 0.5*(-3.0*UW(IJK) + 4.0*UW(IJKP) - UW(IJKPP))
      ENDIF
      IJK = LK(NKM) + LI(I) + J
      IJKM = IJK - NIJ
      IJKMM = IJKM - NIJ
      IF(BCFT(IJ).NE.6) THEN
      DUU3(IJK) = 0.5*(U2(IJKMM) - 4.0*U2(IJKM) + 3.0*U2(IJK))
      DVV3(IJK) = 0.5*(V2(IJKMM) - 4.0*V2(IJKM) + 3.0*V2(IJK))
      DWW3(IJK) = 0.5*(W2(IJKMM) - 4.0*W2(IJKM) + 3.0*W2(IJK))
      DUV3(IJK) = 0.5*(UV(IJKMM) - 4.0*UV(IJKM) + 3.0*UV(IJK))
      DVW3(IJK) = 0.5*(VW(IJKMM) - 4.0*VW(IJKM) + 3.0*VW(IJK))
      DUW3(IJK) = 0.5*(UW(IJKMM) - 4.0*UW(IJKM) + 3.0*UW(IJK))
      ENDIF
 34   CONTINUE
C
C FOR PERIODICITY ALONG THE SOUTH AND NORTH BOUNDARY
C
      DO 35 I=2,NIM
      II = (I-1)*NK
      DO 35 K=2,NKM
      IK = II + K
      IF(BCFS(IK).NE.5) GO TO 35
      IJK = LK(K) + LI(I) + 2
      IJMK = LK(K) + LI(I) + NJM
      DUU2(IJK) = 0.5*(U2(IJK+1) - U2(IJMK))
      DVV2(IJK) = 0.5*(V2(IJK+1) - V2(IJMK))
      DWW2(IJK) = 0.5*(W2(IJK+1) - W2(IJMK))
      DUV2(IJK) = 0.5*(UV(IJK+1) - UV(IJMK))
      DVW2(IJK) = 0.5*(VW(IJK+1) - VW(IJMK))
      DUW2(IJK) = 0.5*(UW(IJK+1) - UW(IJMK))
      IJK = LK(K) + LI(I) + NJM
      IJPK = LK(K) + LI(I) + 2
```

```
      DUU2(IJK) = 0.5*(U2(IJPK) - U2(IJK-1))
      DVV2(IJK) = 0.5*(V2(IJPK) - V2(IJK-1))
      DWW2(IJK) = 0.5*(W2(IJPK) - W2(IJK-1))
      DUV2(IJK) = 0.5*(UV(IJPK) - UV(IJK-1))
      DVW2(IJK) = 0.5*(VW(IJPK) - VW(IJK-1))
      DUW2(IJK) = 0.5*(UW(IJPK) - UW(IJK-1))
   35 CONTINUE
C
      DO 50 K=2,NKM
      DO 50 I=2,NIM
      LIK = LI(I) + LK(K)
      DO 50 J=2,NJM
C
      IJK = LIK + J
      IPJK = IJK + NJ
      IMJK = IJK - NJ
      IJKP = IJK + NIJ
      IJKM = IJK - NIJ
      IJMK=IJK-1
C
      DXKS=QTR*(X(IJK)+X(IJMK)+X(IJKM)+X(IJKM-1)-X(IMJK)-X(IMJK-1)-
     *                X(IJKM-NJ)-X(IJKM-NJ-1))
      DXET=QTR*(X(IJK)+X(IJKM)+X(IMJK)+X(IJKM-NJ)-
     *                X(IJMK)-X(IMJK-1)-X(IJKM-1)-X(IJKM-NJ-1))
      DXZD=QTR*(X(IJK)+X(IJMK)+X(IMJK)+X(IMJK-1)-X(IJKM)-
     *                X(IJKM-NJ)-X(IJKM-1)-X(IJKM-NJ-1))
      DYKS=QTR*(Y(IJK)+Y(IJMK)+Y(IJKM)+Y(IJKM-1)-Y(IMJK)-Y(IMJK-1)-
     *                Y(IJKM-NJ)-Y(IJKM-NJ-1))
      DYET=QTR*(Y(IJK)+Y(IJKM)+Y(IMJK)+Y(IJKM-NJ)-
     *                Y(IJMK)-Y(IMJK-1)-Y(IJKM-1)-Y(IJKM-NJ-1))
      DYZD=QTR*(Y(IJK)+Y(IJMK)+Y(IMJK)+Y(IMJK-1)-Y(IJKM)-
     *                Y(IJKM-NJ)-Y(IJKM-1)-Y(IJKM-NJ-1))
      DZKS=QTR*(Z(IJK)+Z(IJMK)+Z(IJKM)+Z(IJKM-1)-Z(IMJK)-Z(IMJK-1)-
     *                Z(IJKM-NJ)-Z(IJKM-NJ-1))
      DZET=QTR*(Z(IJK)+Z(IJKM)+Z(IMJK)+Z(IJKM-NJ)-
     *                Z(IJMK)-Z(IMJK-1)-Z(IJKM-1)-Z(IJKM-NJ-1))
      DZZD=QTR*(Z(IJK)+Z(IJMK)+Z(IMJK)+Z(IMJK-1)-Z(IJKM)-
     *                Z(IJKM-NJ)-Z(IJKM-1)-Z(IJKM-NJ-1))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C
C Calculate stress gradient source terms
      SUASM(IJK)=-B11*DUU1(IJK)-B21*DUU2(IJK)-B31*DUU3(IJK)
     &          -B12*DUV1(IJK)-B22*DUV2(IJK)-B32*DUV3(IJK)
     &          -B13*DUW1(IJK)-B23*DUW2(IJK)-B33*DUW3(IJK)
      SVASM(IJK)=-B11*DUV1(IJK)-B21*DUV2(IJK)-B31*DUV3(IJK)
     &          -B12*DVV1(IJK)-B22*DVV2(IJK)-B32*DVV3(IJK)
     &          -B13*DVW1(IJK)-B23*DVW2(IJK)-B33*DVW3(IJK)
      SWASM(IJK)=-B11*DUW1(IJK)-B21*DUW2(IJK)-B31*DUW3(IJK)
     &          -B12*DVW1(IJK)-B22*DVW2(IJK)-B32*DVW3(IJK)
     &          -B13*DWW1(IJK)-B23*DWW2(IJK)-B33*DWW3(IJK)
   50 CONTINUE
C
      RETURN
      END
C
C*****************************************************************
      SUBROUTINE CALUIUJ(NIM,NJM,NKM,LK,LI,U2,V2,W2, UV,UW,VW,
     &  GENTB,GENTT,GENTS,GENTN,GENTW,GENTE,  GEN,
     &  BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,
     &  X,Y,Z, FX,FY,FZ, U,V,W, VOL,
     &  TE,ED,VIS, OMX,OMY,OMZ, DENSIT, VISCOS)
C*****************************************************************
```



```
C
      INCLUDE   'param.h'
      INCLUDE   'asmmod.h'
C
      DIMENSION LK(JZ),LI(JX),
     &          X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),
     &          FX(JXYZ),FY(JXYZ),FZ(JXYZ)
      DIMENSION U(JXYZ),V(JXYZ),W(JXYZ),
     &          TE(JXYZ),ED(JXYZ),VIS(JXYZ)
      DIMENSION U2(JXYZ),V2(JXYZ),W2(JXYZ),
     &          UV(JXYZ),UW(JXYZ),VW(JXYZ)
      DIMENSION GEN(JXYZ)
      DIMENSION GENTB(JXY),GENTT(JXY),GENTS(JXZ),GENTN(JXZ),
     &          GENTW(JYZ),GENTE(JYZ)
      DIMENSION BCFW(JYZ),BCFE(JYZ),BCFS(JXZ),BCFN(JXZ),
     &          BCFB(JXY),BCFT(JXY)
C
      INTEGER   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
      DIMENSION A(6,6),B(6)
C
C---CALCULATE ALGEBRAIC STRESS EQUATIONS IN THE FORM
C
C-    Ai1*U2(IJK) + Ai2*V2(IJK) + Ai3*W2(IJK) +
C-    Ai4*UV(IJK) + Ai5*VW(IJK) + Ai6*(UW(IJK) = Bi
C
      DO 10 K=2,NKM
      LKK=LK(K)
      DO 10 I=2,NIM
      LII=LI(I)
      DO 10 J=2,NJM
      IJ=LII+J
      IJK=LKK+IJ
C
      EDK=ED(IJK)/(TE(IJK)+SMALL)
      AUX=(1.-C2ASM)/(C1ASM*ED(IJK)+GEN(IJK)-ED(IJK))
      AUX=1./(AUX*TE(IJK)+SMALL)
C--U2-EQUATION
      A(1,1)=1.5*AUX+2.*DUDX(IJK)
      A(1,2)=-DVDY(IJK)
      A(1,3)=-DWDZ(IJK)
      A(1,4)=2.*DUDY(IJK)-DVDX(IJK)-6.*COMEGA*OMZ
      A(1,5)=-DVDZ(IJK)-DWDY(IJK)
      A(1,6)=2.*DUDZ(IJK)-DWDX(IJK)+6.*COMEGA*OMY
C--V2-EQUATION
      A(2,1)=-DUDX(IJK)
      A(2,2)=1.5*AUX+2.*DVDY(IJK)
      A(2,3)=-DWDZ(IJK)
      A(2,4)=2.*DVDX(IJK)-DUDY(IJK)+6.*COMEGA*OMZ
      A(2,5)=2.*DVDZ(IJK)-DWDY(IJK)-6.*COMEGA*OMX
      A(2,6)=-DUDZ(IJK)-DWDX(IJK)
C--W2-EQUATION
      A(3,1)=-DUDX(IJK)
      A(3,2)=-DVDY(IJK)
      A(3,3)=1.5*AUX+2.*DWDZ(IJK)
      A(3,4)=-DUDY(IJK)-DVDX(IJK)
      A(3,5)=-DVDZ(IJK)+2.*DWDY(IJK)+6.*COMEGA*OMX
      A(3,6)=-DUDZ(IJK)+2.*DWDX(IJK)-6.*COMEGA*OMY
C--UV-EQUATION
      A(4,1)=DVDX(IJK)+2.*COMEGA*OMZ
      A(4,2)=DUDY(IJK)-2.*COMEGA*OMZ
      A(4,3)=0.0
      A(4,4)=AUX+DUDX(IJK)+DVDY(IJK)
      A(4,5)=DUDZ(IJK)+2.*COMEGA*OMY
      A(4,6)=DVDZ(IJK)-2.*COMEGA*OMX
C--VW-EQUATION
      A(5,1)=0.0
      A(5,2)=DWDY(IJK)+2.*COMEGA*OMX
      A(5,3)=DVDZ(IJK)-2.*COMEGA*OMX
      A(5,4)=DWDX(IJK)-2.*COMEGA*OMY
      A(5,5)=AUX+DVDY(IJK)+DWDZ(IJK)
      A(5,6)=DVDX(IJK)+2.*COMEGA*OMZ
```

```
C--UW-EQUATION
        A(6,1)=DWDX(IJK)-2.*COMEGA*OMY
        A(6,2)=0.0
        A(6,3)=DUDZ(IJK)+2.*COMEGA*OMY
        A(6,4)=DWDY(IJK)+2.*COMEGA*OMX
        A(6,5)=DUDY(IJK)-2.*COMEGA*OMZ
        A(6,6)=AUX+DUDX(IJK)+DWDZ(IJK)
C
C--RIGHT HAND SIDE OF EQUATION
        B(1)=AUX*TE(IJK)
        B(2)=AUX*TE(IJK)
        B(3)=AUX*TE(IJK)
        B(4)=0.0
        B(5)=0.0
        B(6)=0.0
        IF(WREFON.EQ.1) THEN
        B1=1./(1.-C2ASM)
        B2=C1P*ED(IJK)/(TE(IJK)+SMALL)
        B3=C2ASM*C2P
C--RHS OF U2-EQUATION
        FW1=B2*(-2.*U2(IJK)*FUNX(IJK)+V2(IJK)*FUNY(IJK)+
       &  W2(IJK)*FUNZ(IJK)-
       &  UV(IJK)*FUNXY(IJK)-UW(IJK)*FUNXZ(IJK)+2.*VW(IJK)*FUNYZ(IJK))
        FW2=B3*(2.*(P11(IJK)-2./3.*GEN(IJK))*FUNX(IJK)-
       &            (P22(IJK)-2./3.*GEN(IJK))*FUNY(IJK)-
       &            (P33(IJK)-2./3.*GEN(IJK))*FUNZ(IJK)+
       &            P12(IJK)*FUNXY(IJK)+P13(IJK)*FUNXZ(IJK)-
       &            2.*P23(IJK)*FUNYZ(IJK))
        B(1)=B(1)+1.5*B1*(FW1+FW2)
C--RHS OF V2-EQUATION
        FW1=B2*(U2(IJK)*FUNX(IJK)-2.*V2(IJK)*FUNY(IJK)+
       & W2(IJK)*FUNZ(IJK)-
       & UV(IJK)*FUNXY(IJK)+2.*UW(IJK)*FUNXZ(IJK)-VW(IJK)*FUNYZ(IJK))
        FW2=B3*(-(P11(IJK)-2./3*GEN(IJK))*FUNX(IJK)+P12(IJK)*FUNXY(IJK)-
       & 2.*P13(IJK)*FUNXZ(IJK)+2.*(P22(IJK)-2./3.*GEN(IJK))*FUNY(IJK)+
       & P23(IJK)*FUNYZ(IJK)-(P33(IJK)-2./3.*GEN(IJK))*FUNZ(IJK))
        B(2)=B(2)+1.5*B1*(FW1+FW2)
C--RHS OF W2-EQUATION
        FW1=B2*(U2(IJK)*FUNX(IJK)+V2(IJK)*FUNY(IJK)-
       &  2.*W2(IJK)*FUNZ(IJK)+
       &  2.*UV(IJK)*FUNXY(IJK)-UW(IJK)*FUNXZ(IJK)-VW(IJK)*FUNYZ(IJK))
        FW2=B3*(-(P11(IJK)-2./3.*GEN(IJK))*FUNX(IJK)-
       & 2.*P12(IJK)*FUNXY(IJK)+
       & P13(IJK)*FUNXZ(IJK)-(P22(IJK)-2./3.*GEN(IJK))*FUNY(IJK)+
       & P33(IJK)*FUNYZ(IJK)+2.*(P33(IJK)-2./3.*GEN(IJK))*FUNZ(IJK))
        B(3)=B(3)+1.5*(FW1+FW2)
C--RHS OF UV-EQUATION
        FW1=-1.5*B2*(UV(IJK)*(FUNX(IJK)+FUNY(IJK))+(U2(IJK)+
       &      V2(IJK))*FUNXY(IJK)+
       &      UW(IJK)*FUNYZ(IJK)+VW(IJK)*FUNXZ(IJK))
        FW2=1.5*B3*((P11(IJK)-2./3.*GEN(IJK))*FUNXY(IJK)+
       &      P12(IJK)*(FUNX(IJK)+FUNY(IJK))+
       &      (P22(IJK)-2./3.*GEN(IJK))*FUNXY(IJK)+P13(IJK)*FUNYZ(IJK)+
       &      P23(IJK)*FUNXZ(IJK))
        B(4)=B(4)+B1*(FW1+FW2)
C--RHS OF VW-EQUATION
        FW1=-1.5*B2*(UW(IJK)*FUNXY(IJK)+UV(IJK)*FUNXZ(IJK)+
       &      VW(IJK)*(FUNY(IJK)+FUNZ(IJK))+V2(IJK)*FUNYZ(IJK)+
       &      W2(IJK)*FUNYZ(IJK))
        FW2=1.5*B3*(P12(IJK)*FUNXZ(IJK)+P13(IJK)*FUNXY(IJK)+
       &      (P22(IJK)-2./3.*GEN(IJK))*FUNYZ(IJK)+
       &      P23(IJK)*(FUNY(IJK)+FUNZ(IJK))+
       &      (P33(IJK)-2./3.*GEN(IJK))*FUNY(IJK)*FUNZ(IJK))
        B(5)=B(5)+B1*(FW1+FW2)
C--RHS OF UW-EQUATION
        FW1=-1.5*B2*(UW(IJK)*(FUNX(IJK)+FUNY(IJK))+U2(IJK)*FUNXZ(IJK)+
       &      W2(IJK)*FUNXZ(IJK)+UV(IJK)*FUNYZ(IJK)+VW(IJK)*FUNXY(IJK))
        FW2=1.5*B3*((P11(IJK)-2./3.*GEN(IJK))*FUNXZ(IJK)+
       &      P13(IJK)*(FUNX(IJK)+FUNZ(IJK))+P12(IJK)*FUNYZ(IJK)+
       &      P23(IJK)*FUNXY(IJK)+(P33(IJK)-2./3.*GEN(IJK))*FUNXZ(IJK))
        B(6)=B(6)+B1*(FW1+FW2)
```



```
C
        ENDIF
C
C-----SOLVE THE 6X6 ALGEBRAIC EQUATIONS
        CALL SOLV(A,B,6)
C
C---N.B. SOLUTION IS IN B(1...6) FOR U2,V2....UW
C
        U2(IJK)=B(1)*RELT+(1.-RELT)*U2(IJK)
        V2(IJK)=B(2)*RELT+(1.-RELT)*V2(IJK)
        W2(IJK)=B(3)*RELT+(1.-RELT)*W2(IJK)
        UV(IJK)=B(4)*RELT+(1.-RELT)*UV(IJK)
        VW(IJK)=B(5)*RELT+(1.-RELT)*VW(IJK)
        UW(IJK)=B(6)*RELT+(1.-RELT)*UW(IJK)
C
C
C--LIMIT NORMAL STRESSES
C
        TAUMAX=2.*TE(IJK)
        TAUMIN=0.0
        U2(IJK)=AMIN1(U2(IJK),TAUMAX)
        U2(IJK)=AMAX1(U2(IJK),TAUMIN)
        V2(IJK)=AMIN1(V2(IJK),TAUMAX)
        V2(IJK)=AMAX1(V2(IJK),TAUMIN)
        W2(IJK)=AMIN1(W2(IJK),TAUMAX)
        W2(IJK)=AMAX1(W2(IJK),TAUMIN)
C
 10     CONTINUE
C
        CALL WALSTRS(NIM,NJM,NKM,LI,LK,FX,FY,FZ,
       & X,Y,Z,VOL, U,V,W, VIS,VISCOS,DENSIT,TE,
       & BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,
       & U2,V2,W2, UV,UW,VW
       & )

C
C--RECALCULATE THE PRODUCTION TERMS
C
        CALL CALPIJ(NIM,NJM,NKM,LI,LK,FX,FY,FZ,
       &  X,Y,Z,VOL, U,V,W, U2,V2,W2, UV,UW,VW,   GEN,
       &  GENTB,GENTT,GENTS,GENTN,GENTW,GENTE,
       &  BCFW,BCFE,BCFS,BCFN,BCFB,BCFT )
C
        RETURN
        END
C
C*****************************************************
        SUBROUTINE WALSTRS(NIM,NJM,NKM,LI,LK,FX,FY,FZ,
       &  X,Y,Z,VOL, U,V,W, VIS,VISCOS,DENSIT,TE,
       &  BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,
       &  U2,V2,W2, UV,UW,VW)
C*****************************************************
C
        INCLUDE   'param.h'
        INCLUDE   'asmmod.h'
C
        DIMENSION LK(JZ),LI(JX),
       &          X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),
       &          FX(JXYZ),FY(JXYZ),FZ(JXYZ)
        DIMENSION U(JXYZ),V(JXYZ),W(JXYZ),
       &          TE(JXYZ),VIS(JXYZ)
        DIMENSION U2(JXYZ),V2(JXYZ),W2(JXYZ),
       &          UV(JXYZ),UW(JXYZ),VW(JXYZ)
        DIMENSION BCFW(JYZ),BCFE(JYZ),BCFS(JXZ),BCFN(JXZ),
       &          BCFB(JXY),BCFT(JXY)
        INTEGER   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
C
C---- CALCULATE MEAN VELOCITY GRADIENTS
C
        CALL PHGRAD(U,DUDX,DUDY,DUDZ,
       &   NIM,NJM,NKM,LI,LK,FX,FY,FZ,
```

```
     &    X,Y,Z,VOL,
     &    BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)
      CALL PHGRAD(V,DVDX,DVDY,DVDZ,
     &    NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &    X,Y,Z,VOL,
     &    BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)
      CALL PHGRAD(W,DWDX,DWDY,DWDZ,
     &    NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &    X,Y,Z,VOL,
     &    BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)
C
C---SET STRESS COMPONENTS AT THE BOUNDARY EDGES.
C
C--WEST & EAST FACE
      DO K=2,NKM
      KK=(K-1)*NJ
      DO J=2,NJM
      JK=KK+J
      IF(BCFW(JK).EQ.4) THEN
      IJK=LK(K)+LI(2)+J
      TAUMAX=2.*TE(IJK)
      TAUMIN=0.0
      VIST=VIS(IJK)-VISCOS
      U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)
      V2(IJK)=-2.*VIST*DVDY(IJK)+2./3.*DENSIT*TE(IJK)
      W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)
      UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))
      VW(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))
      UW(IJK)=-VIST*(DUDZ(IJK)+DWDX(IJK))
      U2(IJK)=AMIN1(U2(IJK),TAUMAX)
      U2(IJK)=AMAX1(U2(IJK),TAUMIN)
      V2(IJK)=AMIN1(V2(IJK),TAUMAX)
      V2(IJK)=AMAX1(V2(IJK),TAUMIN)
      W2(IJK)=AMIN1(W2(IJK),TAUMAX)
      W2(IJK)=AMAX1(W2(IJK),TAUMIN)
      ENDIF
      IF(BCFE(JK).EQ.4) THEN
      IJK=LK(K)+LI(NIM)+J
      TAUMAX=2.*TE(IJK)
      TAUMIN=0.0
      VIST=VIS(IJK)-VISCOS
      U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)
      V2(IJK)=-2.*VIST*DVDY(IJK)+2./3.*DENSIT*TE(IJK)
      W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)
      UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))
      VW(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))
      UW(IJK)=-VIST*(DUDZ(IJK)+DWDX(IJK))
      U2(IJK)=AMIN1(U2(IJK),TAUMAX)
      U2(IJK)=AMAX1(U2(IJK),TAUMIN)
      V2(IJK)=AMIN1(V2(IJK),TAUMAX)
      V2(IJK)=AMAX1(V2(IJK),TAUMIN)
      W2(IJK)=AMIN1(W2(IJK),TAUMAX)
      W2(IJK)=AMAX1(W2(IJK),TAUMIN)
      ENDIF
      ENDDO
      ENDDO
C
C--SOUTH & NORTH FACE
      DO I=2,NIM
      II=(I-1)*NK
      DO K=2,NKM
      IK=II+K
      IF(BCFS(IK).EQ.4) THEN
      IJK=LK(K)+LI(I)+2
      TAUMAX=2.*TE(IJK)
      TAUMIN=0.0
      VIST=VIS(IJK)-VISCOS
      U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)
      V2(IJK)=-2.*VIST*DVDY(IJK)+2./3.*DENSIT*TE(IJK)
      W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)
      UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))
```



```
      VW(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))
      UW(IJK)=-VIST*(DUDZ(IJK)+DWDX(IJK))
      U2(IJK)=AMIN1(U2(IJK),TAUMAX)
      U2(IJK)=AMAX1(U2(IJK),TAUMIN)
      V2(IJK)=AMIN1(V2(IJK),TAUMAX)
      V2(IJK)=AMAX1(V2(IJK),TAUMIN)
      W2(IJK)=AMIN1(W2(IJK),TAUMAX)
      W2(IJK)=AMAX1(W2(IJK),TAUMIN)
      ENDIF
      IF(BCFN(IK).EQ.4) THEN
      IJK=LK(K)+LI(I)+NJM
      TAUMAX=2.*TE(IJK)
      TAUMIN=0.0
      VIST=VIS(IJK)-VISCOS
      U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)
      V2(IJK)=-2.*VIST*DVDY(IJK)+2./3.*DENSIT*TE(IJK)
      W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)
      UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))
      VW(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))
      UW(IJK)=-VIST*(DUDZ(IJK)+DWDX(IJK))
      U2(IJK)=AMIN1(U2(IJK),TAUMAX)
      U2(IJK)=AMAX1(U2(IJK),TAUMIN)
      V2(IJK)=AMIN1(V2(IJK),TAUMAX)
      V2(IJK)=AMAX1(V2(IJK),TAUMIN)
      W2(IJK)=AMIN1(W2(IJK),TAUMAX)
      W2(IJK)=AMAX1(W2(IJK),TAUMIN)
      ENDIF
      ENDDO
      ENDDO
C--BOTTOM & TOP FACE
      DO I=2,NIM
      II=(I-1)*NJ
      DO J=2,NJM
      IJ=LI(I)+J
      IF(BCFB(IJ).EQ.4) THEN
      IJK=LK(2)+LI(I)+J
      TAUMAX=2.*TE(IJK)
      TAUMIN=0.0
      VIST=VIS(IJK)-VISCOS
      U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)
      V2(IJK)=-2.*VIST*DVDY(IJK)+2./3.*DENSIT*TE(IJK)
      W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)
      UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))
      VW(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))
      UW(IJK)=-VIST*(DUDZ(IJK)+DWDX(IJK))
      U2(IJK)=AMIN1(U2(IJK),TAUMAX)
      U2(IJK)=AMAX1(U2(IJK),TAUMIN)
      V2(IJK)=AMIN1(V2(IJK),TAUMAX)
      V2(IJK)=AMAX1(V2(IJK),TAUMIN)
      W2(IJK)=AMIN1(W2(IJK),TAUMAX)
      W2(IJK)=AMAX1(W2(IJK),TAUMIN)
      ENDIF
      IF(BCFT(IJ).EQ.4) THEN
      IJK=LK(NKM)+LI(I)+J
      TAUMAX=2.*TE(IJK)
      TAUMIN=0.0
      VIST=VIS(IJK)-VISCOS
      U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)
      V2(IJK)=-2.*VIST*DVDY(IJK)+2./3.*DENSIT*TE(IJK)
      W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)
      UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))
      VW(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))
      UW(IJK)=-VIST*(DUDZ(IJK)+DWDX(IJK))
      U2(IJK)=AMIN1(U2(IJK),TAUMAX)
      U2(IJK)=AMAX1(U2(IJK),TAUMIN)
      V2(IJK)=AMIN1(V2(IJK),TAUMAX)
      V2(IJK)=AMAX1(V2(IJK),TAUMIN)
      W2(IJK)=AMIN1(W2(IJK),TAUMAX)
      W2(IJK)=AMAX1(W2(IJK),TAUMIN)
      ENDIF
      ENDDO
```

```
      ENDDO
C
      RETURN
      END
C
C****************************************************
      SUBROUTINE SOLV(A,BB,N)
C****************************************************
C.....A GAUSS ELIMINATION SOLVER
C
      DIMENSION A(N,N),B(10),C(10),BB(N),X(10),MME(10)
C
      EP =1.E-19
      DO 10 J=1,N
10    MME(J)=J
      DO 20 I=1,N
      Y=0.
      DO 30 J=I,N
      IF(ABS(A(I,J)).LT.ABS(Y)) GOTO 30
      K=J
      Y=A(I,J)
30    CONTINUE
C
      IF(ABS(Y).LT.EP) THEN
      WRITE(*,*)
      WRITE(9,*)
      DO 35 IA=1,N
      WRITE(*,1000) (A(IA,JA),JA=1,N)
35    CONTINUE
      PRINT*,'THERE IS NO CONVERSE MATRIX'
      STOP 2222
      ENDIF
C
      Y=1./Y
      DO 40 J=1,N
      C(J)=A(J,K)
      A(J,K)=A(J,I)
      A(J,I)=-C(J)*Y
      B(J)=A(I,J)*Y
40    A(I,J)=A(I,J)*Y
      A(I,I)=Y
      J=MME(I)
      MME(I)=MME(K)
      MME(K)=J
      DO 11 K=1,N
      IF(K.EQ.I) GOTO 11
      DO 12 J=1,N
      IF(J.EQ.I) GOTO 12
      A(K,J)=A(K,J)-B(J)*C(K)
12    CONTINUE
11    CONTINUE
20    CONTINUE
      DO 33 I=1,N
      DO 44 K=1,N
      IF(MME(K).EQ.I) GOTO 55
44    CONTINUE
55    IF(K.EQ.I) GOTO 33
      DO 66 J=1,N
      W=A(I,J)
      A(I,J)=A(K,J)
66    A(K,J)=W
      IW=MME(I)
      MME(I)=MME(K)
      MME(K)=IW
33    CONTINUE
1000  FORMAT(4X,1P5E13.4)
      DO 50 I=1,N
      X(I)=0.
      DO 50 J=1,N
50    X(I)=X(I)+A(I,J)*BB(J)
      DO 60 I=1,N
```



```
60    BB(I)=X(I)
      RETURN
      END
C
C****************************************************
      SUBROUTINE CALPIJ(NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &  X,Y,Z,VOL, U,V,W, U2,V2,W2, UV,UW,VW, GEN,
     &  GENTB,GENTT,GENTS,GENTN,GENTW,GENTE,
     &  BCFW,BCFE,BCFS,BCFN,BCFB,BCFT )
C****************************************************
C
      INCLUDE   'param.h'
      INCLUDE   'asmmod.h'
C
      DIMENSION LK(JZ),LI(JX),
     &          X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),
     &          FX(JXYZ),FY(JXYZ),FZ(JXYZ)
      DIMENSION U(JXYZ),V(JXYZ),W(JXYZ)
      DIMENSION U2(JXYZ),V2(JXYZ),W2(JXYZ),
     &          UV(JXYZ),UW(JXYZ),VW(JXYZ)
      DIMENSION GEN(JXYZ)
      DIMENSION BCFW(JYZ),BCFE(JYZ),BCFS(JXZ),BCFN(JXZ),
     &          BCFB(JXY),BCFT(JXY)
      DIMENSION GENTB(JXY),GENTT(JXY),GENTS(JXZ),GENTN(JXZ),
     &          GENTW(JYZ),GENTE(JYZ)
      INTEGER   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
C
C---- CALCULATE MEAN VELOCITY GRADIENTS
C
      CALL PHGRAD(U,DUDX,DUDY,DUDZ,
     &   NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &   X,Y,Z,VOL,
     &   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)
      CALL PHGRAD(V,DVDX,DVDY,DVDZ,
     &   NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &   X,Y,Z,VOL,
     &   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)
      CALL PHGRAD(W,DWDX,DWDY,DWDZ,
     &   NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &   X,Y,Z,VOL,
     &   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)
C
      DO K=2,NKM
      DO I=2,NIM
      DO J=2,NJM
      IJK=LK(K)+LI(I)+J
C
      P11(IJK)=-2.*(U2(IJK)*DUDX(IJK)+UV(IJK)*DUDY(IJK)+
     &              UW(IJK)*DUDZ(IJK))
      P22(IJK)=-2.*(UV(IJK)*DVDX(IJK)+V2(IJK)*DVDY(IJK)+
     &              VW(IJK)*DVDZ(IJK))
      P33(IJK)=-2.*(UW(IJK)*DWDX(IJK)+VW(IJK)*DWDY(IJK)+
     &              W2(IJK)*DWDZ(IJK))
      P12(IJK)=-(U2(IJK)*DVDX(IJK)+V2(IJK)*DUDY(IJK)+
     &           UV(IJK)*DWDZ(IJK)+
     &           UW(IJK)*DVDZ(IJK)+VW(IJK)*DUDZ(IJK))
      P13(IJK)=-(U2(IJK)*DWDX(IJK)+W2(IJK)*DUDZ(IJK)+
     &           UV(IJK)*DWDY(IJK)+
     &           UW(IJK)*DVDY(IJK)+VW(IJK)*DUDY(IJK))
      P23(IJK)=-(V2(IJK)*DWDY(IJK)+W2(IJK)*DVDZ(IJK)+
     &           UV(IJK)*DWDX(IJK)+
     &           UW(IJK)*DVDX(IJK)+VW(IJK)*DUDX(IJK))
C
      GEN(IJK)=HAF*ABS(P11(IJK)+P22(IJK)+P33(IJK))
C
      ENDDO
      ENDDO
      ENDDO
C
C--- MODIFY GENERATION TERM CLOSE TO A WALL
C
```

```
C------- TOP & BOTTOM WALL
      DO I=2,NIM
      DO J=2,NJM
      IJ=LI(I)+J
C--BOTTOM BOUNDARY
      IJK=LK(2)+IJ
      IF(BCFB(IJ).EQ.4) GEN(IJK)=GENTB(IJ)
C--TOP BOUNDARY
      IJK=LK(NKM)+IJ
      IF(BCFT(IJ).EQ.4) GEN(IJK)=GENTT(IJ)
      ENDDO
      ENDDO
C
C------- SOUTH & NORTH BOUNDARIES
C
      DO I=2,NIM
      II=(I-1)*NK
      DO K=2,NKM
C------- SOUTH BOUNDARY
      IK=II+K
      IJK=LK(K)+LI(I)+2
      IF(BCFS(IK).EQ.4) GEN(IJK)=GENTS(IK)
C------- NORTH BOUNDARY
      IJK=LK(K)+LI(I)+NJM
      IF(BCFN(IK).EQ.4) GEN(IJK)=GENTN(IK)
      ENDDO
      ENDDO
C
C------- EAST & WEST BOUNDARIES
C
      DO K=2,NKM
      KK=(K-1)*NJ
      DO J=2,NJM
C------- WEST BOUNDARY
      JK=KK+J
      IJK=LK(K)+LI(2)+J
      IF(BCFW(JK).EQ.4) GEN(IJK)=GENTW(JK)
C------- EAST BOUNDARY
      IJK=LK(K)+LI(NIM)+J
      IF(BCFE(JK).EQ.4) GEN(IJK)=GENTE(JK)
      ENDDO
      ENDDO
C
      RETURN
      END
C
C**********************************************
      SUBROUTINE PHGRAD(PHI,DPHDX,DPHDY,DPHDZ,
     &   NIM,NJM,NKM,LI,LK,FX,FY,FZ,
     &   X,Y,Z,VOL,
     &   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)
C
C**********************************************
C
      INCLUDE   'param.h'
      INCLUDE   'asmmod.h'
C
      DIMENSION PHI(JXYZ),DPHDX(JXYZ),DPHDY(JXYZ),DPHDZ(JXYZ)
      DIMENSION LK(JZ),LI(JX),
     &          X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),
     &          FX(JXYZ),FY(JXYZ),FZ(JXYZ)
      DIMENSION BCFW(JYZ),BCFE(JYZ),BCFS(JXZ),BCFN(JXZ),
     &          BCFB(JXY),BCFT(JXY)
C
      INTEGER   BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
C
C-- WEST BOUNDARY CELL FACE
C
      DO 10 K=2,NKM
      LKK=LK(K)
      KKJ=(K-1)*NJ
```



```
      KKI=(K-1)*NI
C
      DO 10 I=2,NIM
      IK=(I-1)*NK+K
      LII=LI(I)
C
      DO 10 J=2,NJM
      IJ=LII+J
      IJK=LKK+IJ
      IJKP=IJK+NIJ
      IJKM=IJK-NIJ
      IPJK=IJK+NJ
      IMJK=IJK-NJ
      IJPK=IJK+1
      IJMK=IJK-1
C
      FXE=FX(IJK)
      JK=KKJ+J
      IF(I.EQ.NIM.AND.BCFE(JK).EQ.6) FXE=0.5
      FXW=1.-FXE
      FYN=FY(IJK)
      IK=(I-1)*NK+K
      IF(J.EQ.NJM.AND.BCFN(IK).EQ.6) FYN=0.5
      FYS=1.-FYN
      FZT=FZ(IJK)
      IJ=LII+J
      IF(K.EQ.NKM.AND.BCFT(IJ).EQ.6) FZT=0.5
      FZB=1.-FZT
C
      DXKS=QTR*(X(IJK)+X(IJMK)+X(IJKM)+X(IJKM-1)-X(IMJK)-X(IMJK-1)-
     *          X(IJKM-NJ)-X(IJKM-NJ-1))
      DXET=QTR*(X(IJK)+X(IJKM)+X(IMJK)+X(IJKM-NJ)-
     *          X(IJMK)-X(IMJK-1)-X(IJKM-1)-X(IJKM-NJ-1))
      DXZD=QTR*(X(IJK)+X(IJMK)+X(IMJK)+X(IMJK-1)-X(IJKM)-
     *          X(IJKM-NJ)-X(IJKM-1)-X(IJKM-NJ-1))
      DYKS=QTR*(Y(IJK)+Y(IJMK)+Y(IJKM)+Y(IJKM-1)-Y(IMJK)-Y(IMJK-1)-
     *          Y(IJKM-NJ)-Y(IJKM-NJ-1))
      DYET=QTR*(Y(IJK)+Y(IJKM)+Y(IMJK)+Y(IJKM-NJ)-
     *          Y(IJMK)-Y(IMJK-1)-Y(IJKM-1)-Y(IJKM-NJ-1))
      DYZD=QTR*(Y(IJK)+Y(IJMK)+Y(IMJK)+Y(IMJK-1)-Y(IJKM)-
     *          Y(IJKM-NJ)-Y(IJKM-1)-Y(IJKM-NJ-1))
      DZKS=QTR*(Z(IJK)+Z(IJMK)+Z(IJKM)+Z(IJKM-1)-Z(IMJK)-Z(IMJK-1)-
     *          Z(IJKM-NJ)-Z(IJKM-NJ-1))
      DZET=QTR*(Z(IJK)+Z(IJKM)+Z(IMJK)+Z(IJKM-NJ)-
     *          Z(IJMK)-Z(IMJK-1)-Z(IJKM-1)-Z(IJKM-NJ-1))
      DZZD=QTR*(Z(IJK)+Z(IJMK)+Z(IMJK)+Z(IMJK-1)-Z(IJKM)-
     *          Z(IJKM-NJ)-Z(IJKM-1)-Z(IJKM-NJ-1))
C......INLINE
      B11=DYET*DZZD-DYZD*DZET
      B12=DXZD*DZET-DXET*DZZD
      B13=DXET*DYZD-DYET*DXZD
      B21=DZKS*DYZD-DZZD*DYKS
      B22=DXKS*DZZD-DXZD*DZKS
      B23=DXZD*DYKS-DXKS*DYZD
      B31=DYKS*DZET-DYET*DZKS
      B32=DZKS*DXET-DZET*DXKS
      B33=DXKS*DYET-DXET*DYKS
C......INLINE
C
      PHEW=PHI(IPJK)*FXE-PHI(IMJK)*(1.-FX(IMJK))+
     &     PHI(IJK)*(FXW-FX(IMJK))
      PHNS=PHI(IJPK)*FYN-PHI(IJMK)*(1.-FY(IJMK))+
     &     PHI(IJK)*(FYS-FY(IJMK))
      PHTB=PHI(IJKP)*FZT-PHI(IJKM)*(1.-FZ(IJKM))+
     &     PHI(IJK)*(FZB-FZ(IJKM))
C
C--- GRADIENTS
C
      DPHDX(IJK)=(PHEW*B11+PHNS*B21+PHTB*B31)/VOL(IJK)
      DPHDY(IJK)=(PHEW*B12+PHNS*B22+PHTB*B32)/VOL(IJK)
      DPHDZ(IJK)=(PHEW*B13+PHNS*B23+PHTB*B33)/VOL(IJK)
```

```
C
10    CONTINUE
C
      RETURN
      END
C
C--------------------------------------------------------
C-- param.h
C
      PARAMETER (JX=102)
      PARAMETER (JY=45)
      PARAMETER (JZ=26)
      PARAMETER (JXY=JX*JY)
      PARAMETER (JXZ=JX*JZ)
      PARAMETER (JYZ=JY*JZ)
      PARAMETER (JXYZ=JX*JY*JZ)
C
      PARAMETER (GREAT=1.0E+30)
      PARAMETER (SMALL=1.0E-30)
      PARAMETER (HAF=0.5)
      PARAMETER (QTR=0.25)
      PARAMETER (AHT=0.125)
      PARAMETER (IEQ=2)

C
C--------------------------------------------------------
C-- asmmod.h
C
C
C  COMMON BLOCKS FOR VARIABLES INSIDE 3D ASM MODULE
C
      COMMON/ASMCB0/
     &           NI,NJ,NK,NIJ,NIK,NJK,NIJK
      COMMON/ASMCB1/
     &           P11(JXYZ),P22(JXYZ),P33(JXYZ),P12(JXYZ),P13(JXYZ),
     &           P23(JXYZ)
      COMMON/ASMCB2/
     &           DUDX(JXYZ),DUDY(JXYZ),DUDZ(JXYZ),DVDX(JXYZ),DVDY(JXYZ),
     &           DVDZ(JXYZ), DWDX(JXYZ),DWDY(JXYZ),DWDZ(JXYZ)
      COMMON/ASMCB3/
     &           FUNX(JXYZ),FUNY(JXYZ),FUNZ(JXYZ),FUNXY(JXYZ),
     &           FUNYZ(JXYZ),FUNXZ(JXYZ)
      COMMON/ASMCB4/
     &           C1ASM,C2ASM,C1P,C2P,WREFON,COMEGA,RELT
C
```

CHAPTER 8

Related Publications and Presentations

During the course of this work, some related turbulence modeling work was published or presented at different meetings. Copies of these papers are listed in this chapter, they include;

(1) A. H. Hadid ad M. M. Sindir "Comparative study of advanced turbulence models for turbomachinery" NACA CP-3174, 1992.

This paper was presented at the advanced Earth-to-Orbit propulsion technology conference held at NASA Marshall Space Flight Center in Huntsville, Alabama on May 19-21, 1992. The work tests different correction to the standard k-ε turbulence models that accounts for streamline curvature and rotations using different near wall treatments.

(2) A. H. Hadid, M. E. DeCroix and M. M. Sindir "Advanced turbulence models for turbomachinery" NASA CP-3221, 1993.

This paper was presented at the eleventh workshop for compuatational fluid dynamics applications in rocket propulsion held at NASA Marshal Space Flight Center in Huntsville, Alabama on April 20-22, 1993. The paper outlined the progress of the 2D/axisymmetric single and multi-scale k-ε turbulence module deck developments.

(3) A. Hadid, M. Sindir, C. Chen and H. Wei "Computations of confined swirling flows with high order turbulence models in a modular form"

This paper was presented at the twelfth workshop for computational fluid dynamics applications in rocket propulsion held at NASA Marshal Space Flight Center in Huntsville, Alabama April-May 1994. The paper presented the status of the 2D/axisymmetric second order closure models using the algebraic and the full Reynolds stress models.

(4) A. H. Hadid, M. M. Sindir and R. I. Issa "A numerical study of two-dimensional vortex shedding from rectangular cylinders" published in the CFD Journal July, 1992.

This paper presents a test for an anisotropic k-ε turbulence model. This model is an improvement on the standard k-ε model since it can predicts Reynolds stress anisotropies without the need to solve additional equations for the stresses.

(5) A. H. Hadid, N. N. Mansour and O. Zeman "Single point modeling of rotating turbulent flows" Proceedings of the 1994 summer program, CTR, NASA Ames/Stanford University.

This paper tests a new one-point closure model that incorporates the effects of rotation on the power-law decay exponent of the turbulent kinetic energy. A modification to the ε-equation proposed by Zeman using large eddy simulation results was used. A new definition of the mean rotation was proposed based on critical point theory to generalize the effects of rotation on turbulence to arbitrary mean deformations.

COMPARATIVE STUDY OF ADVANCED TURBULENCE MODELS FOR TURBOMACHINERY

A. H. Hadid and M. M. Sindir
CFD Technology Center
Rocketdyne Division, Rockwell International
Mail Code IB39, 6633 Canoga Avenue
Canoga Park, CA 91303

ABSTRACT

Development and assessment of the standard k–ε turbulence model for rotating flows with different near-wall treatments is presented. These include the standard wall function[1], Patel's two-layer model[2], and Lam and Bremhorst[3] low-Reynolds number model. Two test cases were chosen to validate these models for rotating flows. The first, from Daily and Nece[4] , is for a rotating disk cavity in which recirculation and secondary flows are induced by the rotating element. The second case is that of a confined double concentric jets with a sudden expansion by Roback and Johnson[5].

It is shown that near-wall effects are important close to rotating walls and that the two-layer model behaves better than the other two near-wall models. For confined swirling flows with fixed walls, the near wall effects are of secondary importance to the Reynold's stress anisotropy.

INTRODUCTION

Accurate predictions of turbulent flows are crucial to the design and analysis of many physical and engineering applications. Increases in available computational capabilities have permitted the development and testing of sophisticated models in the numerical simulation of turbulent flows. Direct numerical simulation, where all essential scales of the turbulent flow are resolved by solving the unsteady Navier-Stokes equations, are possible only at low to moderate Reynolds numbers. Turbulent flow analysis for engineering applications, therefore, can only be achieved by utilizing the time-averaged Navier-Stokes equations coupled with some level of modelling.

The complex structure of swirling flowfields requires careful consideration of the turbulence model derivation and development. The analysis of turbulent transport and modeling evolves from the Reynolds-averaged Navier-Stokes equations and auxiliary equations for velocity and length scales for eddy viscosity specifications. Simple eddy viscosity models based on the Boussinesq hypothesis of linear relationship between turbulent shear stress and rate of strain have been quite successful in predicting a wide variety of turbulent flows.

One of the widely used models is the two-equation k-ε model. The model developed originally by Launder and Spalding[1] was successful in providing good predictions for a large range of turbulent flows. The equations can be derived from the full transport equations for the Reynolds stresses assuming fully turbulent flow. Effects such as that of rotation which are included in the Reynolds stress equations are cancelled out and the resulting scalar k-ε equations are invariant to system rotation.

For low-Reynolds number flows close to solid boundaries, adjustments to the model are needed to bridge the viscous dominated sublayer region with the fully turbulent flow region. The success of the wall function method depends on the universality of the turbulent structure near the wall. In many complex flows, however, the flowfield near the wall has to be determined accurately and the traditional wall-function method is not satisfactory. This is because the specification of all turbulence quantities in terms of the friction velocity fail at separation where the flow near the wall is no longer controlled by the wall shear stress. Patel et al.[6] assessed the relative performance of various models which describe the near-wall flows and found that there are still areas of improvements needed to accurately model flow behavior near the wall.

Jones and Launder[7] extended the original k-ε model to the low-Reynolds number form which allowed the calculation to be performed all the way to the wall. Numerical difficulties of accurately resolving the large gradients close to the wall necessitates resolving the wall region with very fine grid structure. Chen and Patel[2] introduced a method to resolve the near-wall region which combines the standard k-ε model with the one-equation model of Wolfshtein[8] near the wall. In this "two-layer" model an algebraically prescribed eddy-viscosity for the wall region is coupled to the k-ε model to describe the details of the flow in the vicinity of the wall. Momentum and continuity equations are solved up to the wall and this reduces the physical uncertainties of near-wall turbulence and the numerical difficulties of resolving the vary large gradients of turbulence parameters.

The purpose of this paper is to discuss the application of the k-ε turbulence model with various near-wall treatments in the prediction of confined swirling flows. These models include, the standard wall function approach (WF), Chen and Patel's[2] two-layer model (2L), and Lam and Bremhorst[3] low-Reynolds number model (LB).

Evaluation of the various turbulence models was performed by comparison with two selected experimental studies. The first is that of Daily and Nece[4] where rotating disk cavity circulation and secondary flows are induced by a rotating wall. The second is that of Roback and Johnson[5] for a confined double concentric jets with a sudden expansion. Flow swirl in this case is induced by imposing a tangential velocity component at the outer jet.

Numerical predictions for turbulent flows in two-dimensional axisymmetric geometries were obtained using a finite-volume second order upwind differencing scheme on a non-staggered grid with a pressure correction method based on the SIMPLE algorithm[9]. The development and evaluation of the turbulence models for rotating flows is part of an ongoing program to assess different models for rotating machinery applications. A discussion on the effects of swirl and streamline curvature on the turbulence structure through the gradient Richardson number formulation is given. Key problem areas will be identified and recommendations for the near-wall treatment as they pertain to rotating flows will be proposed.

MODEL AND EQUATION FORMULATION

Consider an incompressible, statistically steady and axisymmetric turbulent flow, the Reynolds averaged momentum and continuity equations can be expressed in a generalized form as;

$$\frac{\partial(\rho\Phi)}{\partial t}+\frac{\partial(\rho u\Phi)}{\partial x}+\frac{1}{r}\frac{\partial(\rho v r\Phi)}{\partial r}=\frac{\partial}{\partial x}(\Gamma_{\Phi x}\frac{\partial\Phi}{\partial x})+\frac{1}{r}\frac{\partial}{\partial r}(r\Gamma_{\Phi r}\frac{\partial\Phi}{\partial r})+S_\Phi \tag{1}$$

where Φ is the dependent variable

Φ= u, v, w for the axial, radial, and tangential velocities

ρ, μ, and S_Φ are the fluid density, viscosity and the source terms for the variable Φ

The source terms for the dependent variables are;

Axial direction, $\Phi=u$, $\Gamma_{\Phi x}=2\mu_e$, $\Gamma_{\Phi r}=\mu_e$

$$S_u=-\frac{\partial p}{\partial x}+\frac{1\partial}{r\partial r}(\mu_e\, r\frac{\partial v}{\partial x}) \tag{2}$$

Radial direction, $\Phi=v$, $\Gamma_{\Phi x}=\mu_e$, $\Gamma_{\Phi r}=2\mu_e$

$$S_v=\frac{\partial}{\partial x}(\mu_e\frac{\partial u}{\partial r})-2\,\mu_e\frac{v}{r^2}+\rho\frac{w^2}{r}-\frac{\partial p}{\partial r} \tag{3}$$

Tangential direction, $\Phi=w$, $\Gamma_{\Phi x}=\mu_e$, $\Gamma_{\Phi r}=\mu_e$

$$S_w=-\frac{\rho v w}{r}-\frac{w}{r^2}\frac{\partial}{\partial r}(r\mu_e) \tag{4}$$

<u>TURBULENCE MODELS</u> In the two-equation k-ε model transport equations for the turbulent kinetic energy (k) and energy dissipation (ε) can be written in the same general form as equation (1).

Turbulent Kinetic energy equation

$$\Phi=k,\ \Gamma_{\Phi x}=\Gamma_{\Phi r}=\mu+\frac{\mu_t}{\sigma_k}\ \text{, and }\ S_\Phi=G-\rho\varepsilon \tag{5}$$

Energy dissipation equation

$$\Phi=\varepsilon,\ \Gamma_{\Phi x}=\Gamma_{\Phi r}=\mu+\frac{\mu_t}{\sigma_\varepsilon},\ \text{and }\ S_\Phi=\frac{\varepsilon}{k}(C_1f_1G-C_2f_2\rho\varepsilon) \tag{6}$$

σ_k and σ_ε are turbulent Schmidt numbers G denotes the rate of production of the turbulent kinetic energy and is express as;

$$G=\mu_e\,\{\,2[\,(\frac{\partial u}{\partial x})^2+(\frac{\partial v}{\partial r})^2+(\frac{v}{r})^2\,]+(\frac{\partial u}{\partial r}+\frac{\partial v}{\partial x})^2+(\frac{\partial w}{\partial x})^2+(\frac{\partial w}{\partial r}-\frac{w}{r})^2\,\} \tag{7}$$

$\mu_e=\mu+\mu_t$, *and the eddy viscosity is obtained from* $\mu_t=C_\mu\, f_\mu\, \rho\,\frac{k^2}{\varepsilon}$

C_μ, C_1, C_2, σ_k and σ_ε are constants whose values are 0.09, 1.44, 1.92, 1.0, 1.0, respectively.

Near-Wall Treatment A near-wall turbulent flow can be divided into two regions, the inner viscous sublayer where low turbulence Reynolds number effects are important and the velocities decrease rapidly to zero at the wall, and an outer fully turbulent region. The successful use of the k-ε turbulence model for many complex flows depends on how accurately the flowfield near the wall is determined. Different models are used to treat this thin sublayer region, they include:

Wall function methods, the following equations are assumed to hold

$$u^+ = y^+ \qquad \text{for } y^+ \le 11.6 \tag{8}$$

$$u^+ = \frac{1}{k}\ln(Ey^+) \qquad \text{for } y^+ > 11.6 \tag{9}$$

where $u^+ = \frac{u}{u_\tau}$, $y^+ = \frac{u_\tau y}{v}$ and $u_\tau = (\frac{\tau_w}{\rho})^{1/2}$

τ_w is the wall shear stress which is estimated from

$$\tau_w = \frac{\mu u_p}{\delta} \qquad \text{for } y^+ \le 11.6 \tag{10}$$

$$\tau_w = \frac{\kappa C_\mu^{0.25} \rho u_p k^{0.5}}{\ln(EC_\mu^{0.25} \rho\delta k^{0.5}/\mu)} \qquad \text{for } y^+ > 11.6 \tag{11}$$

where u_p denotes the velocity component parallel to the wall, and δ is the normal distance from the wall

In this approach, k and ε equations are solved with $f_\mu = f_1 = f_2 = 1$ only in the fully turbulent region beyond some distance from the wall. Boundary conditions, i.e., velocity components and turbulence parameters at that distance are specified in terms of the friction velocity (u_τ).

In the low-Reynolds number model, the flow is resolved all the way to the wall with a very fine mesh. Many models have been proposed that are based on the k-ε model and differ mainly in the choice of the damping functions f_μ, f_1 and f_2 to bridge the gap between the sublayer and the fully turbulent regions. Lam and Bremhorst's model[3] is used in this work, where

$$f_\mu = [\, 1 - \exp(-0.016R_y)\,]^{1/2} (1 + \frac{20.5}{R_t})$$

$$f_1 = 1 + (\frac{0.06}{f_\mu})^3 \quad \text{and} \quad f_2 = 1 - \exp(-R_t^2)$$

$R_y = \frac{k^{1/2}y}{v}$ and $R_t = \frac{k^2}{v\varepsilon}$ are turbulent Reynolds numbers

These damping functions tend to unity with increasing distance from the wall. In order to resolve the very large gradients of turbulence parameters a fine mesh is required in the viscous sublayer which increases the computational time and numerical difficulties may be encountered.

In order to alleviate some of the problems encountered in the low-Reynolds number approach and yet accurately resolve the near-wall region, Chen and Patel[2] pursued the two-layer concept. In this model a simple algebraically prescribed eddy-viscosity model for the wall region is coupled to the k-ε model for the outer flow to describe the details of the flow. Unlike the low-Reynolds number model that requires the solution of transport equations of both k and ε all the way to the wall, the one-equation model requires the solution of only the turbulent kinetic energy equation in the sublayer region while algebraically specifying the eddy-viscosity and energy dissipation.

$$v_t = C_\mu k^{1/2} L_\mu \quad \text{and} \quad \varepsilon = k^{3/2}/L_\varepsilon$$

The length scales L_μ and L_ε contain the necessary damping effects in the near-wall region in terms of the turbulence Reynolds number R_y

$$L_\mu = C_l\, y\, [\, 1 - \exp(-R_y/A_\mu)\,] \tag{12}$$

$$L_\varepsilon = C_l\, y\, [\, 1 - \exp(-R_y/A_\varepsilon)\,] \tag{13}$$

The length scales L_μ and L_ε become linear and approach $C_l\, y$ with increasing distance form the wall. $C_l = \kappa\, C_\mu^{-0.75}$ with $A_\varepsilon = 2C_l$. Chen and Patel[2] gave values for the constant $A_\mu = 70$. The damping effects decay rapidly with distance from the wall independent of the magnitude of the wall shear stress. The matching between the one-equation and the standard k-ε models is carried out along prescribed grid lines where $R_y \sim 200$.

STREAMLINE CURVATURE AND SWIRL CORRECTIONS Turbulent flows in many engineering applications such as turbomachinery and combustion devices are frequently subjected to complicating influences such as mean strain and body forces due to rotation. In such complex flows streamline curvature and swirl can exert a large influence on the structure of turbulence. Bradshaw[10] reviewed the effects of streamline curvature and discussed the large effect exerted on shear-flow turbulence by curvature of streamlines in the plane of the main shear. So and Mellor[11] suggested that the appropriate parameter governing this effect is $F = \frac{u/R}{\partial u/\partial y}$, where R is the radius of streamline curvature. Militzer et al.[12] provided a simple generalization of this parameter for a 2-D recirculating flow as

$$F = \frac{(u^2+v^2)^{1/2}/R}{(\partial u/\partial y + \partial v/\partial x)} \qquad (14)$$

They modified the turbulence production term G in the turbulent energy equations to include curvature effects and obtained improved predictions. Launder et al[13] proposed a simple modification to the constant C_2 in the ε-equation to account for streamline curvature due to swirl in the form

$$C_2 = 1.0 - 0.2\, R_{i_s} \qquad (15)$$

where R_{i_s} is a swirl Richardson number defined by

$$R_{i_s} = \frac{w/r^2\, \partial(rw)/\partial r}{(\partial u/\partial r)^2 + (r\, \partial(w/r)/\partial r)^2} \qquad (16)$$

Another expression of R_{i_s} can also be derived as

$$R_{i_s} = \frac{k^2}{\varepsilon^2}\frac{w}{r^2}\frac{\partial(rw)}{\partial r} \qquad (17)$$

The basis of the above correction is that the effect of swirl on turbulence can be modelled through an increase in the length scale of the energetic turbulence eddies.

Abujela and Lilley[14] used a modified C_2 form (Eq. 15) with both definitions of R_{i_s} from Equations (16) and (17) as applied to turbulent swirling flows. They concluded that Eq. (16) Richardson number gave better comparisons with experiment as compared to Eq. (17) Richardson's number. They also found the value of C_2 obtained from Eq. (15) had to be limited to $0.1 \leq C_2 \leq 2.4$ with C_μ and other constants assigned their conventional values.

Srinivasan and Mongia[15] further split the Richardson number into two parts - the swirl Richardson number R_{i_s} and the curvature Richardson number R_{i_c} and corrected C_2 in the ε-equations as:

$$C_2 = 1.92 \exp(2\alpha_s R_{i_s} + 2\alpha_c R_{i_c}) \qquad (18)$$

where R_{i_s} is given by equations (16) or (17) and

$$R_{i_c} = \frac{(u^2+v^2)^{1/2}/R}{\left(\frac{1}{r}\frac{\partial ur}{\partial r} + \frac{\partial v}{\partial x}\right)} \qquad (19)$$

where R is the radius of curvature given by $R = \frac{(u^2+v^2)^{3/2}}{uv\left(\frac{1}{r}\frac{\partial rv}{\partial r} - \frac{\partial u}{\partial x}\right)}$, and α_s and α_c are constants with values ranging between 0.1 and 2.4.

Chang et al[16] investigated the streamline curvature effects in the k-ε model. They managed to obtain satisfactory results in their hybrid k-ε model where modifications of curvature effects in C_2 is made only in regions where the streamline curvature is large.

In the present study curvature and swirl modifications are made to C_2 similar to Eq. (18) of Srinivasan and Mongia with R_{i_s} as in Eq. (16) and R_{i_c} as in Eq. (19). The exponential form ensures that C_2 will never become negative. Numerical testing with several values of α_s and α_c reveal that C_2 may become very large and therefore, had to be limited to $0.1 \leq C_2 \leq 2.4$.

MODEL EVALUATION

The various near-wall treatment models are analyzed by comparing model predictions with experimental data. Two cases of rotating flow experiments were selected for validation, they include; Daily and Nece[4] for rotating disk cavity experiment and Roback and Johnson[5] for swirling flow in a confined double concentric jets with a sudden expansion. The main criterion for selecting these cases is the different mechanisms used to generate swirling flows. In Daily and Nece experiment flow rotation is induced by the rotating wall, while in Roback and Johnson's Experiment, swirl is imparted to the flow by an outer swirling jet into a sudden expansion. Different rotation mechanisms affecting turbulence can highlight the differences between various turbulence models and offer certain corrections that would prove useful in accurately analyzing the effects of swirl.

CASE (1) - DAILY AND NECE[4] In their experimental and analytic study Daily and Nece[4] analyzed the steady-state turbulent flow in enclosed rotating disk cavities. They characterized the existence of four flow regimes depending on the rotational Reynolds number and cavity aspect ratio. The two-dimensional axisymmetric flow considered is that of an incompressible flow bounded by a disk (rotor) and a stationary end wall (stator) of a chamber as shown in Figure 1. The ratio of the axial clearance between the rotor and the stator (s) to the radius of the disk (a) is 0.0255. The disk rotates with a rotational Reynolds number $R=4.4\times10^6$ defined as $R=\Omega a^2/\nu$, where Ω is the disk rotational speed in rad/sec and ν is the kinematic viscosity.

Numerical computations were performed on a 33x75 grid with different grid clustering near the walls for the different near-wall models. Figure 2, shows the velocity vectors at the top region of the cavity using the WF model. Centrifugal forces move the fluid radially outward on the disk, axially away from the disk on the wall casing, and radially inwards on the stationary end wall. Figure 3, shows the axial variations of the radial velocity component (v) at a radial position r/a=0.765. The agreement is fair with some discrepancy for all near-wall models close to the rotating disk. Figure 4, shows the axial variation of the tangential velocity (w) component at the radial position. At the rotating disk (x=0), the tangential velocity component approach the value ($a\Omega$). The 2L near-wall model seem to offer closer agreement with the data than the other two models.

The presence of corner regions presents a difficulty in defining the normal distances used in the definition of turbulent Reynolds number (R_y). In the present analysis, values of the normal distance from a wall were based on the normal distance to the nearest solid boundary. Streamline curvature and swirl corrections have not been used in this case.

CASE (2) - ROBACK AND JOHNSON[5] Predictions of the experiments of Roback and Johnson[5] have been presented by several workers, e.g. Sloan et al.[17] and Durst and Wennergerg[18]. Unfortunately, inlet profiles were not provided in their experiment. Therefore, calculations were started at the expansion plane using the measured velocity profiles at 5mm downstream of the expansion after some adjustments near the edges of the coaxial jets. Measurements of all three main turbulent intensities were used to calculate inlet values of the turbulent kinetic energy. Energy dissipation rate was estimated from

$$\varepsilon = \frac{C_\mu k^{3/2}}{L} \tag{20}$$

where L is a length scale of turbulence at the inlet of the order of $L=10^{-4}$ m.

Figure 5, shows an illustration of the test chamber geometry. The confluence plane of the primary (inner) and secondary (outer) jet streams coincides with the chamber expansion plane. The chamber diameter is about twice the secondary tube diameter. The exit from the 8-bladed, 30°, free vortex swirl generator is located approximately 0.05 m upstream from the confluence plane.

A prevalent phenomenon in axisymmetric swirling flows in such geometries is the "bubble" or vortex breakdown which has been studied extensively[19,20,21,22]. The near axisymmetric breakdown can be partially understood from a simplified analysis of the role of pressure and centrifugal forces. It is identified by a slowly varying vortex core which undergoes an abrupt and rapid deceleration, forming a free stagnation point, followed by a region of flow reversal. It is known that the structure of vortex breakdown is unstable and asymmetric in the azimuthal direction, and displays unsteadiness in the axial direction[23,24]. However, no periodic or nonaxisymmetric behavior attributable to the vortex breakdown was observed in Roback and Johnson's experiment.

In the numerical simulation of the experiment, a 150x100 grid nodes was used with different clustering on the walls for the different near-wall models used. Figure 6, shows the velocity vectors indicating the presence of a closed recirculation zone at the center with additional zones at the corner downstream and between the inner jet and the outward diverted secondary jet. The figure also shows flow diversion outwards with high gradients characterized by large turbulent shear and fluctuation levels.

Comparisons were made of the radial variations of flow variables at two axial locations, x=0.025 m upstream of the vortex bubble and x=0.102 m located inside the vortex bubble. Figure 7a, shows the radial variation of the axial velocity profile at x=0.025 m using the WF method, 2L model and LB model. Fair agreement is predicted by the different models. They also seem to predict small negative velocities at a radial position r~0.0153 m (the interface between the inner and outer jets), slightly underpredicting its strength and width. Figure 7b, shows the radial variation of the axial velocity profiles at x=0.102 m. The 2L model shows a better agreement with the experimental data. These velocities are slightly underpredicted above the outer jet diameter.

Radial variations of the tangential velocities are shown in Figure 8a and 8b at x=0.025 m and x=0.102 m respectively. Figure 8a shows that the 2L model offers a better agreement with the experiment as compared with the WF and LB models. At x=0.102 m, Figure 8b shows that the swirl velocity is underpredicted. That is because the radial transfer of circumferential velocity is highly dependent on the turbulent diffusion mechanisms which are not accurately modelled in the isotropic eddy-viscosity k-ε model used here.

The turbulent intensity predictions for the k-ε model using the different near-wall treatments seem to follow similar trends as shown in Figures 9(a,b), 10(a,b), and, 11(a,b). In general within the approximations of the isotropic k-ε model, the 2L model offer a marginal improvements over the WF and LB near wall models. The peaks in the axial, radial, and, tangential turbulence intensities occur around the edges of the inner and outer jets. Figure 13a, shows the axial-azimuthal Reynolds stress profile at x=0.025 m. Figure 12b and 12c, show the axial-radial and radial-azimuthal Reynolds stress profiles at x=0.102 m.

The analysis of the main turbulent intensities and of the Reynolds stress components using the isotropic eddy-viscosity k-ε turbulent model do not reveal exclusively the advantage of one near-wall model over the other. Moreover, Reynolds shear stress profiles are sensitive to the upstream inlet conditions and the developing mean flowfield. Although the mean flow quantities show a general trend of improved predictions using the 2L near-wall model, the main effects of turbulence are due to anisotropy of Reynolds stresses especially around the highly sheared region of the outward diverted outer jet and the vortex bubble.

Streamline curvature and swirl corrections have been attempted in the present analysis with little success. Corrections of C_2 using equation (18) with equations (16) and (19) for the swirl and curvature Richardson numbers. Figure 13(a,b) show the radial distribution of the radial velocity at x=0.025m, and the axial velocity at x=0.102m. Small improvement is detected with these corrections. The constants α_s and α_c used are those recommended by Srinivasan and Mongia[15] in their calculations (α_S=-0.75 and α_C=-2.0). These constants were not optimized in the present calculations.

CONCLUSIONS

Flow predictions were performed for the standard k-ε turbulence model with different near-wall treatments to assess their performance when applied to rotating flows. Comparisons of predictions with the experimental data of Daily & Nece, and Roback & Johnson show reasonable agreement for all near-wall models and in general, the two-layer model seem to offer better comparisons compared to the wall function and Lam & Bremhorst low-Reynolds number models. From a computational perspective, the two-layer model require less computer time and relatively few grid points in the wall region than the low-Reynolds number model and is less sensitive to the location of the interface between the sublayer and the fully turbulent regions. Streamline curvature and swirl corrections show small improvements. However, further study is needed to optimize their constants.

REFERENCES

(1) Launder, B. E, and Spalding, D. B. "The Numerical Computation of Turbulent Flows", Computer Methods in Applied Mechanics and Engineering, Vol. 3, 1974, pp. 269-289

(2) Chen, H. C. and Patel, V. C. "Near-Wall Turbulence Models for Complex Flows Including Separation", AIAA Journal, vol. 26, No. 6, 1988, pp. 641-648

(3) Lam, C. and Bremhorst, K. "A Modified Form of the k-ε Model for Predicting Wall Turbulence", Trans. ASME J. Fluids Eng., Vol. 103, 1981, pp. 456-460

(4) Daily, J. W. and Nece, R. E. "Chamber Dimension Effects on Induced Flow and Frictional Resistance of Enclosed Rotating Disks", Trans. ASME J. Basic Eng., 1960, pp. 217-232

(5) Roback R., and Johnson, B. "Mass and Momentum Turbulent Transport Experiment With Confined Swirling Co-Axial Jets", NASA CR-168252, 1983

(6) Patel, V. C., Rodi, W., and Scheuerer, G. "Turbulence Models for Near-Wall and Low Reynolds Number Flows: A Review", AIAA Journal, Vol. 23, No. 9, 1985, pp. 1308-1319

(7) Jones, W. P., and Launder, B. E. "The Calculation of Low-Reynolds-Number Phenomena With a Two-Equation Model of Turbulence", Int. J. Heat & Mass Transfer, vol.16, 1973, pp. 1119-1130

(8) Wolfshtein, M. "The Velocity and Temperature Distribution in One-Dimensional Flow With Turbulence Augmentation and Pressure Gradients" Int. J. Heat & Mass Transfer, vol. 12, 1969, pp. 301-318

(9) Patankar, S. V., "Numerical Heat Transfer and Fluid FLow", McGraw-Hill, New York, 1980
(10) Bradshaw, p. "Effects of Streamline Curvature on Turbulent Flow" AGARDograph No. 169, 1973
(11) So, R. and Mellor, G. "Experiments on Convex Curvature Effects in Turbulent Boundary Layers", J. Fluid Mech., Vol. 60, 1973, pp. 43-62
(12) Militzer, J., Nicoll, W., and Alpay, S. "Some Observations on the Numerical Calculations of the Recirculation Region of Twin Parallel Symmetric Jet Flow", Symp. on Turbulent Shear Flows, University Park, Pennsylvania, April 18-20, 1977
(13) Launder, B., Priddin, C., and Sharma, B. "The Calculation of Turbulent Boundary Layer on Spinning and Curved Surfaces", ASME J. Fluids Eng. 1977, pp. 231-239
(14) Abujela, M. T., and Lilley, D. G. "Limitations and Empirical Extensions of the k-ε Model as Applied to Turbulent Confined Swirling Flows", Paper AIAA-84-0441, Reno, Nevada, Jan. 9-12, 1984
(15) Srinivasan, R., and Mongia, H. "Numerical Computation of Swirling Recirculating Flows: Final Report" NASA CR-165196, 1980
(16) Chang, K. C., Chen, C. S., and Uang, C. I. "A Hybrid k-ε Turbulence Model of Recirculating Flow" Int. J. for Numerical Methods in Fluids, Vol. 12, 1991, pp. 369-382
(17) Sloan, D. G., Smith, P. J., and Smoot, L. D. "Modelling of Swirl in Turbulent Flow Systems" Prog. Energy Combust. Sci., Vol. 12, 1986, pp. 163-250
(18) Durst, F., and Wennerberg, D. "Numerical Aspects of Calculation of Confined Swirling Flows With Internal Recirculation" Int. J. for Numerical Methods in Fluids, Vol. 12, 1991, pp. 203-224
(19) So., K. L. "Vortex Phenomena in a Conical Diffuser" AIAA J., Vol. 5, 1967, pp. 1072-1078
(20) Garg, A. K., and Leibovich, S. "Spectral Characteristics of Vortex Breakdown Flowfields", Physics Fluids, Vol. 22, 1979, pp. 2053-2064
(21) Faler, J. H., and Leibovich, S. "Disrupted States of Vortex Flow and Vortex Breakdown", Physics Fluids, Vol. 20, 1977, pp. 1385-1400
(22) Escudier, M. P. and Zehnder, N. "Vortex Flow Regimes" J. Fluid Mech., Vol. 115, 1982, pp. 105-121
(23) Sarpkaya, T. "On Stationary and Travelling Vortex Breakdowns" J. Fluid Mech., Vol.45, 1971, pp.545-559
(24) Faler, J. H., and Leibovich, S. "An Experimental Map of the Internal Structure of a Vortex Breakdown" J. Fluid Mech., Vol. 86, 1978, pp. 313-335

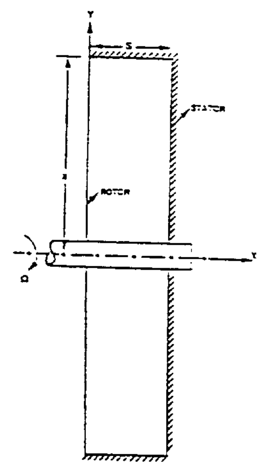


Fig. 1 Rotating Closed Cavity

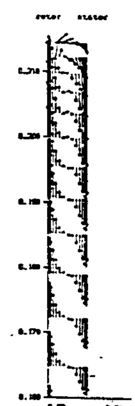


Fig. 2 Velocity Vectors

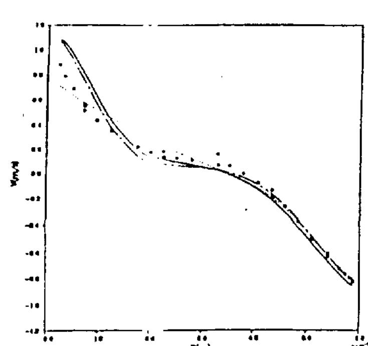


Fig. 3 Axial Distribution of Radial Velocity (m/s), at r/a=0.765

Dally et al
Wall Function Method
Two-Layer Model
Low-Reynolds Number Model

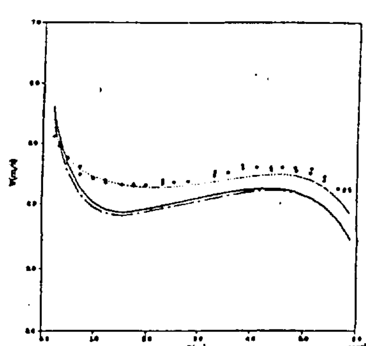


Fig. 4 Axial Distribution of Tangential Velocity (m/s), at r/a=0.765

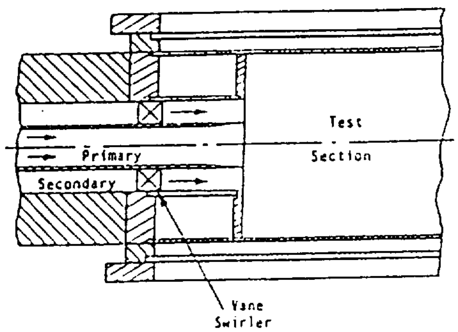


Fig. 5 Swirling Coaxial Jets Discharging into an Expanded Duct

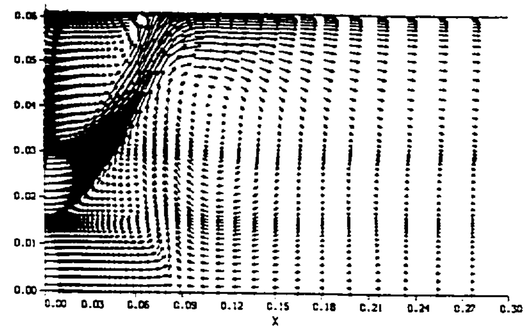


Fig. 6 Velocity Vectors

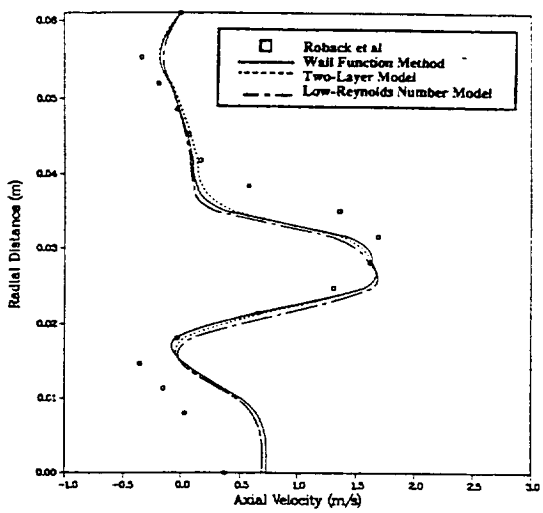


Fig. 7a Radial Distribution of Axial Velocity (m/s) at x-0.025m

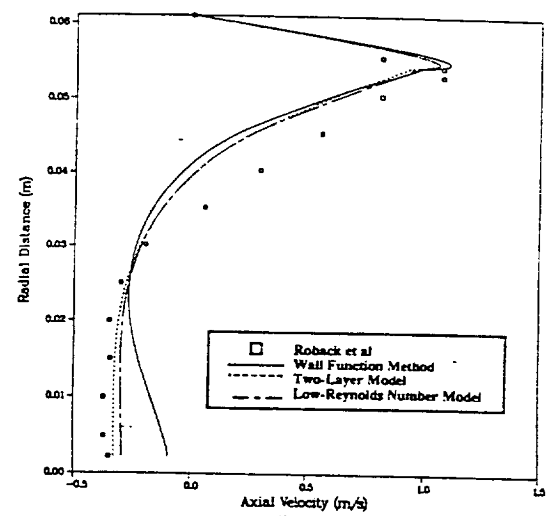


Fig. 7b Radial Distribution of Axial Vellocity (m/s) at x-0.102m

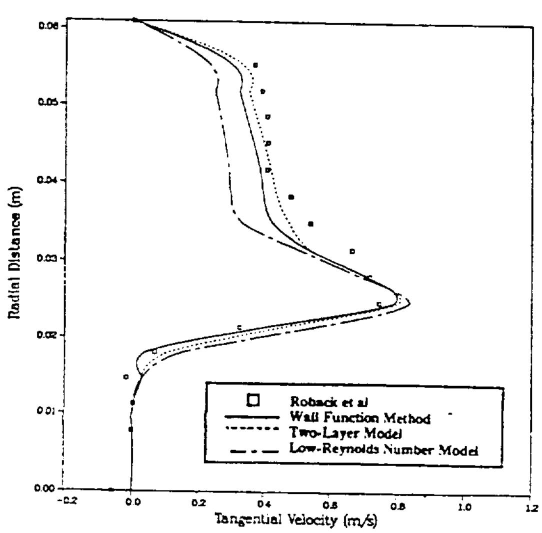


Fig. 8a Radial Distribution of Tangential Velocity (m/s) at x-0.025m

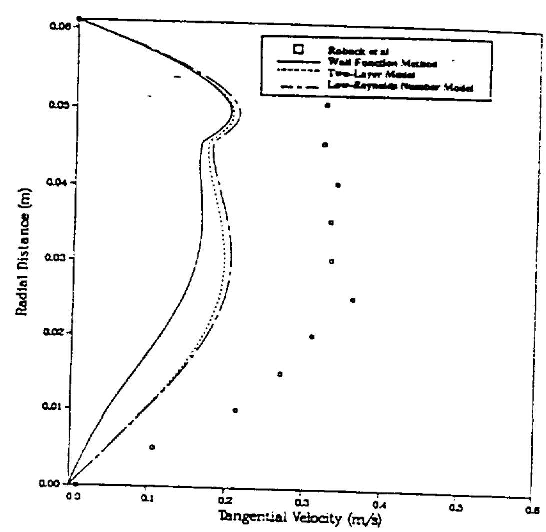


Fig. 8b Radial Distribution of Tangential Velocity (m/s) at x-0.102m

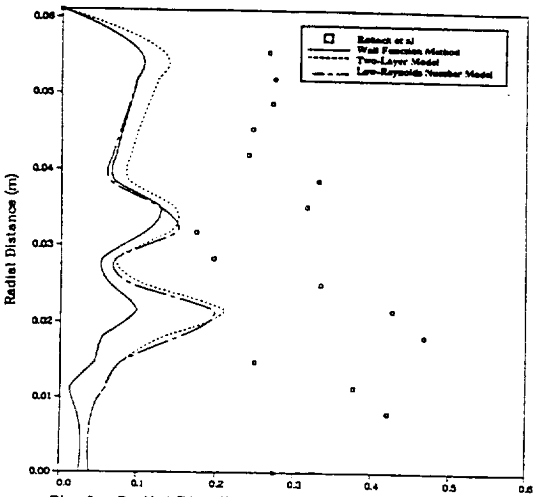


Fig. 9a Radial Distribution of Axial Turbulence Intensity at x-0.025m

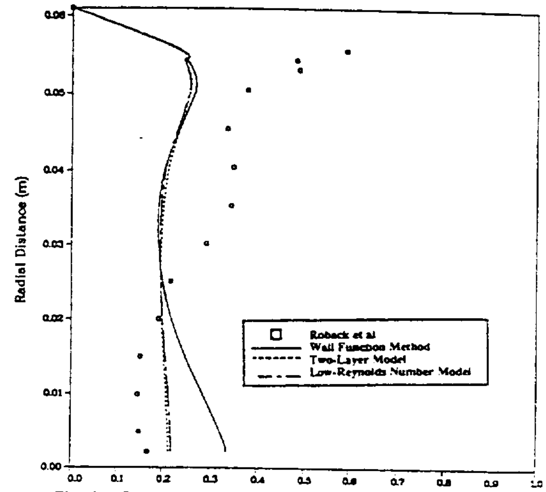


Fig. 9b Radial Distribution of Axial Turbulence Intensity at x-0.102m

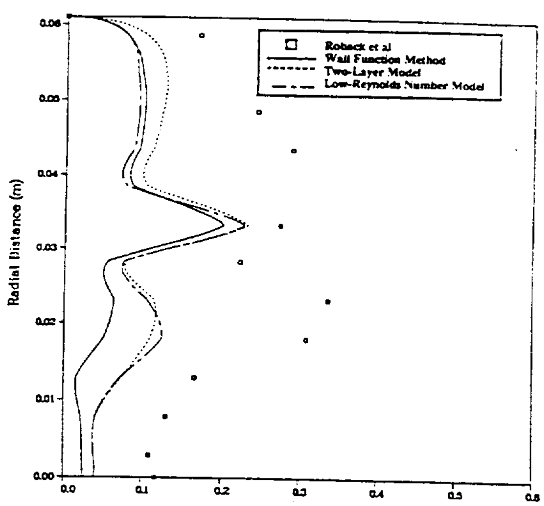


Fig. 10a Radial Distribution of Radial Turbulence Intensity at x-0.025m

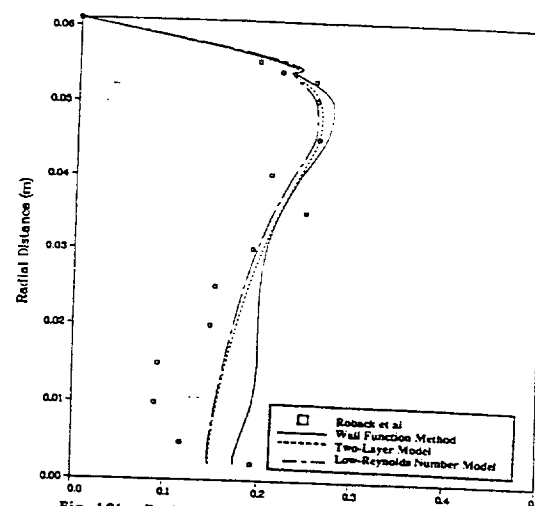


Fig. 10b Radial Distribution of Radial Turbulence Intensity at x-0.102m

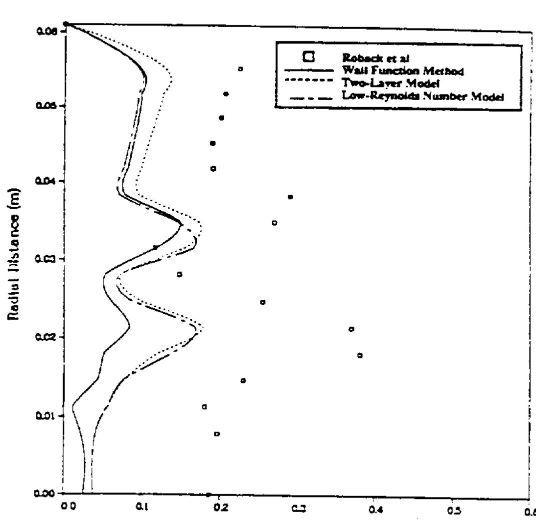


Fig. 11a Radial Distribution of Tangential Turbulence Intensity at x-0.025m

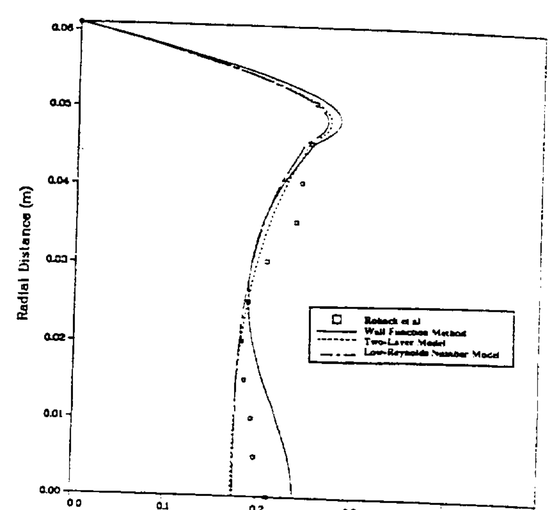


Fig. 11b Radial Distribution of TangentialTurbulence Intensity at x-0.102m

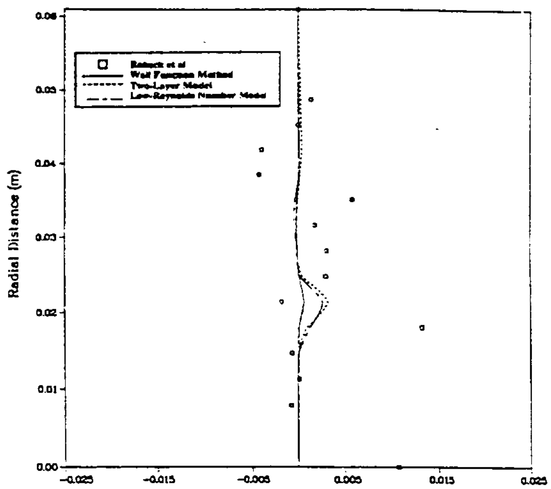


Fig. 12a Radial Distribution of Axial-Azimuthal Reynolds Stresse at x-0.025m

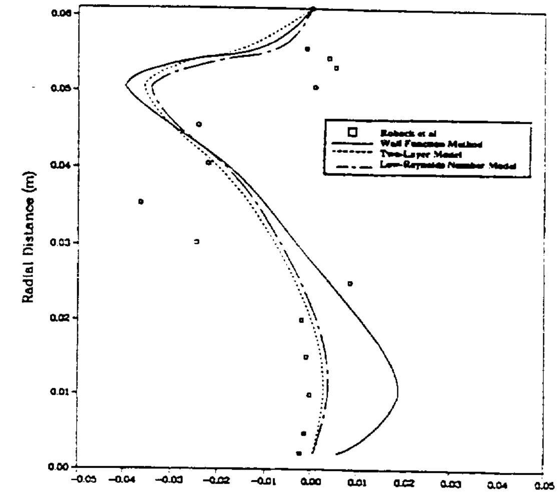


Fig. 12b Radial Distribution of Axial-Radial Reynolds Stresse at x-0.102m

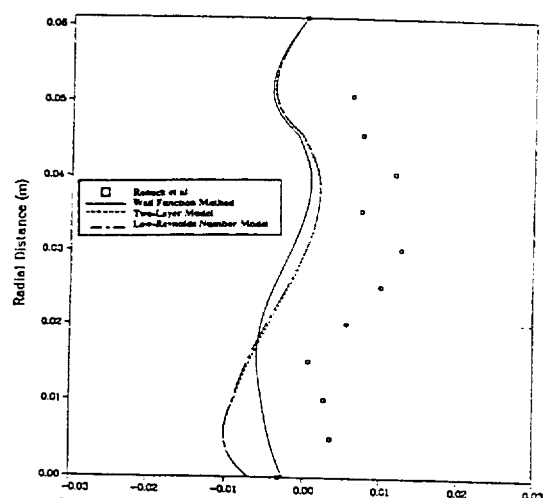


Fig. 12c Radial Distribution of Radial-Azimuthal Reynolds Stress at x-0.102m

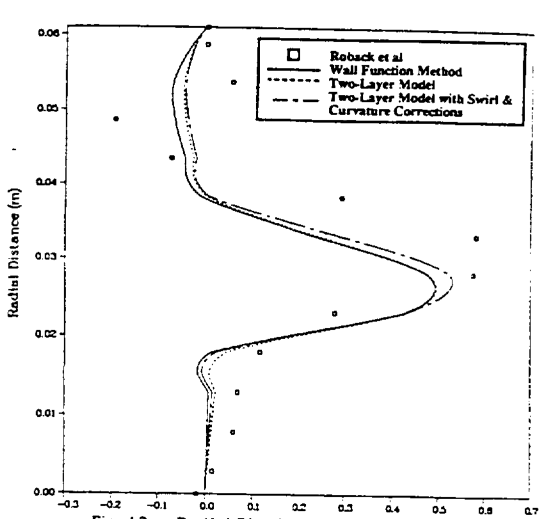


Fig. 13a Radial Distribution of Radial Velocity at x-0.025m

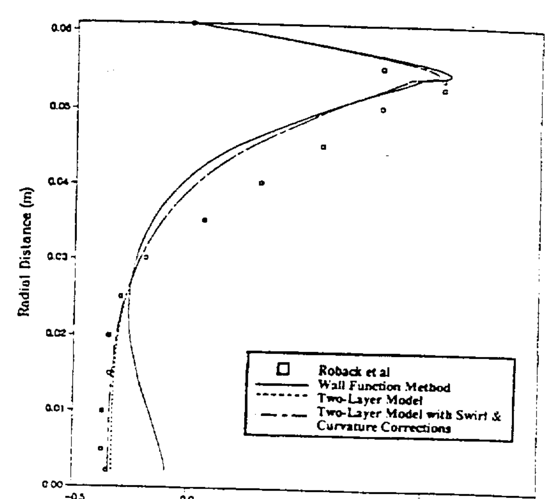


Fig. 13b Radial Distribution of Axial Velocity at x-0.102m

ADVANCED TURBULENCE MODELS FOR TURBOMACHINERY

Ali H. Hadid, Michele E. DeCroix, and Munir M. Sindir
Rocketdyne Division, Rockwell International

ABSTRACT

Development and assessment of the single-time-scale k-ε turbulence model with different near-wall treatments and the multi-scale k-ε turbulence model for rotating flows are presented. These turbulence models are coded as self-contained module decks that can be interfaced with a number of CFD main flow solvers. For each model, a stand-alone module deck with its own formulation, discretization scheme, solver and boundary condition implementations is presented. These satellite decks will take as input (from a main flow solver) the velocity field, grid, boundary condition specifications and will deliver turbulent quantities as output. These modules were tested as a separate entities and although many logical and programming problems were overcome only wider use and further testing can render the modules sufficiently "fool proof".

DEVELOPMENT OF A MODULAR FORMAT FOR GENERAL USE TURBULENCE MODELS

Ali H. Hadid
Michele E. DeCroix

Rockwell International, Rocketdyne Division

Workshop for Computational Fluid Dynamic Applications in Rocket Propulsion

April 20-22, 1993
NASA Marshall Space Flight Center

Rockwell International
Rocketdyne Division

CFD 93-013-001/D1/AHH

STRUCTURE OF FUTURE CODES LEADS TO DEVELOPMENT OF MODULES

- **FUTURE CODES COMPOSED AT TIME OF EXECUTION**
  - INTELLIGENT DRIVERS
  - MODULES FOR PHYSICAL MODELS

- **INTEGRAL PRE- AND POSTPROCESSING TOOLS**

- **COMMON DATA BASE**

Rockwell International
Rocketdyne Division

CFD 93-013-028/D1/AHH

TURBULENCE MODELS TO BE ASSESSED

PHENOMENOLOGICAL SINGLE POINT CLOSURE MODELS

- SINGLE-SCALE
  - 2-EQUATION MODELS K- ε (SKEM)
  - ALGEBRAIC STRESS MODELS (ASM)
  - REYNOLDS STRESS MODELS (RSM)
- MULTI-SCALE
  - 2-EQUATION MODELS K- ε (MKEM)

• THE 2-D/AXISYMMETRIC SINGLE-SCALE K-ε MODULE DECK (KEMOD-1) AND THE 2-D/AXISYMMETRIC MULTI-SCALE K-ε MODULE DECK (KEMOD-2) ARE COMPLETE

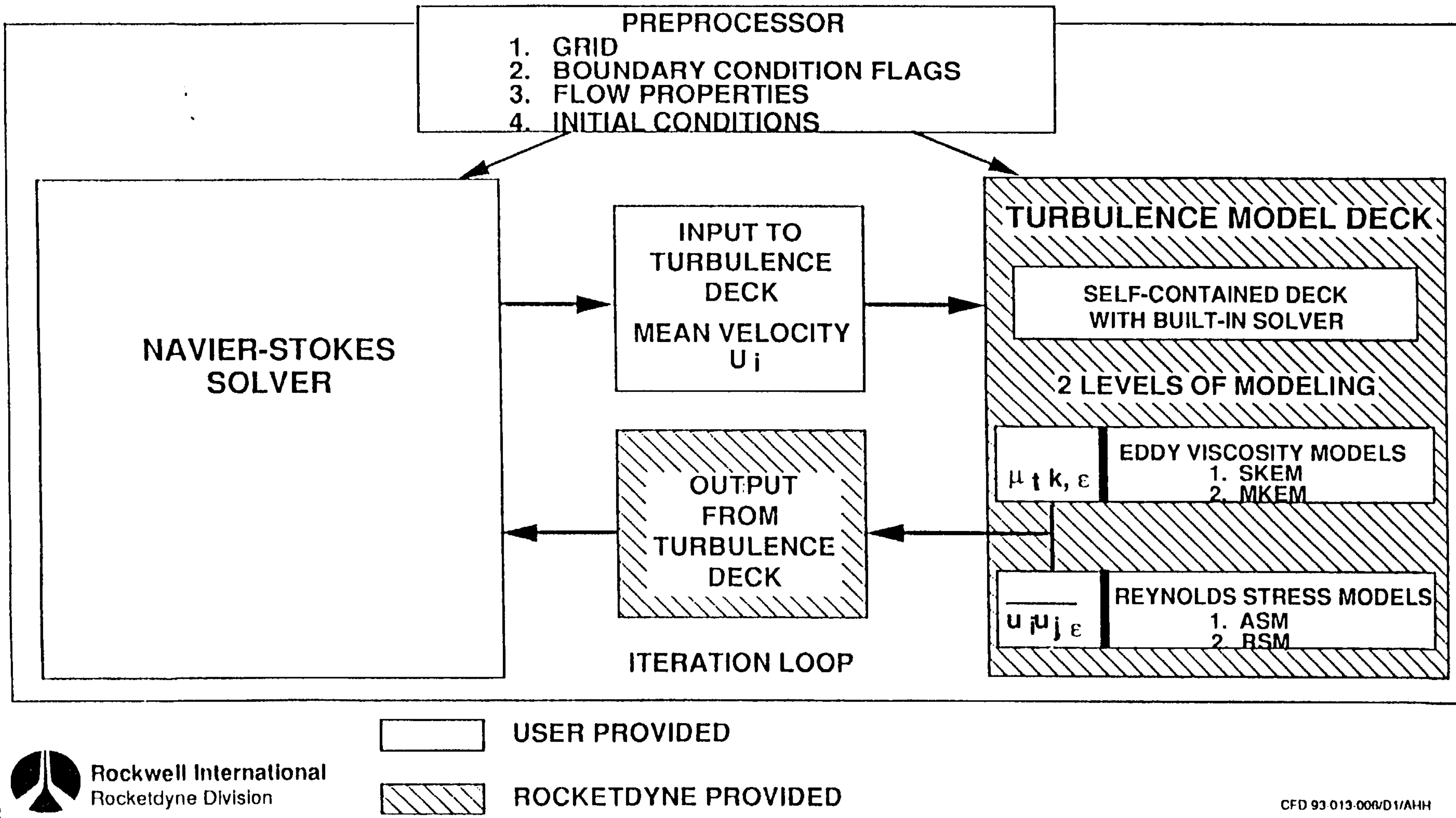
• NEAR-WALL TREATMENTS WILL INCLUDE (WHERE APPROPRIATE) WALL FUNCTIONS, MULTILAYER MODELS, AND LOW-REYNOLDS NUMBER APPROXIMATIONS

Rockwell International
Rocketdyne Division

CFD 93-013-005/D1/AHH

TURBULENCE MODEL DECK STRUCTURE AND INTEGRATION WITH NAVIER-STOKES SOLVER

- MODULES ARE BASED ON THE PHENOMENOLOGICAL SINGLE POINT TURBULENCE CLOSURE MODELS
- THEY ARE STRUCTURED BASICALLY TO ACCEPT AS INPUT THE MEAN FLOW VELOCITIES FROM A N-S SOLVER AND TO RETURN TURBULENCE QUANTITIES TO THE SOLVER



Rockwell International
Rocketdyne Division

CFD 93.013.000/D1/AHH

SINGLE-SCALE k-ε MODEL

- GENERALIZED TRANSPORT EQUATION IN 2-D/AXISYMMETRIC GEOMETRY

$$\frac{\partial(\rho u \phi)}{\partial x} + \frac{1}{r}\frac{\partial}{\partial r}(\rho r v \phi) = \frac{\partial}{\partial x}\left(\Gamma_{\phi x}\frac{\partial \phi}{\partial x}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r \Gamma_{\phi r}\frac{\partial \phi}{\partial r}\right) + S_\phi$$

- U-MOMENTUM $\quad \phi = u, \; \Gamma_{\phi x} = 2\mu_e, \; \Gamma_{\phi r} = \mu_e$

$$S_u = -\frac{\partial P}{\partial x} + \frac{1}{r}\frac{\partial}{\partial r}\left(\mu_e r \frac{\partial v}{\partial x}\right)$$

- V-MOMENTUM $\quad \phi = v, \; \Gamma_{\phi x} = \mu_e, \; \Gamma_{\phi r} = 2\mu_e$

$$S_v = -\frac{\partial P}{\partial x} - \frac{\partial}{\partial r}\left(\mu_e r \frac{\partial u}{\partial x}\right) - 2\mu_e \frac{v}{r^2} + \frac{\rho w^2}{r}$$

SINGLE-SCALE k-ε MODEL (CONT'D)

- W-MOMENTUM

$$\phi = w, \quad \Gamma_{\phi x} = \mu_e, \quad \Gamma_{\phi r} = \mu_e$$

$$S_w = -\frac{\rho v w}{r} + \frac{w}{r^2}\frac{\partial}{\partial r}(r\mu_e)$$

- TURB. KINETIC ENERGY

$$\phi = K, \quad \Gamma_{\phi x} = \mu + \frac{\mu_t}{\sigma_k} = \Gamma_{\phi r}$$

$$S_\phi = G - \rho\varepsilon$$

- TURB. ENERGY DISSIPATION

$$\phi = \varepsilon, \quad \Gamma_{\phi x} = \mu + \frac{\mu_t}{\sigma_k} = \Gamma_{\phi r}$$

$$S_\phi = \frac{\varepsilon}{k}(c_1 f_1 G - c_2 f_2 \rho\varepsilon)$$

Rockwell International
Rocketdyne Division

CFD 93 013-008/D1/AHH

SINGLE-SCALE k-ε MODEL (CONT'D)

- $G = \mu_e \left\{ 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2 + \left(\frac{v}{r}\right)^2 \right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial r} - \frac{w}{r}\right)^2 \right\}$

- $\mu_t = C_\mu f_\mu \rho \frac{k^2}{\varepsilon}$

  $\mu_e = \mu + \mu_t$

  $C_\mu = 0.09, \; c_1 = 1.44, \; c_2 = 1.92, \; \sigma_k = 1.0, \; \sigma_\varepsilon = 1.0$

Rockwell International
Rocketdyne Division

CFD 93 013-008A/D1/AHH

KEMOD-1 MODULE DECK

- KEMOD-1 IS THE SINGLE-SCALE K-ε TURBULENCE MODULE DECK. IT CONSISTS OF TWO SEPARATE ROUTINES KEMOD AND MODIFY WHICH HAVE TO BE LINKED TO THE MAIN FLOW SOLVER
- KEMOD IS CALLED WITHIN THE ITERATION LOOP OF THE MAIN FLOW SOLVER
- THE MEAN VELOCITIES AND OTHER VARIABLES ARE PASSED TO THE MODULE THROUGH ITS ARGUMENT LIST (EXPLAINED IN THE USER'S MANUAL)
- A NONSTAGGERED BODY FITTED GRID ARRANGEMENT IS USED BY THE MODULE. IT USES THE MEAN FLOW VARIABLES (VELOCITIES AND MASS FLUXES) TO CONSTRUCT THE DISCRETIZED ALGEBRAIC EQUATION
- DISCRETIZED ALGEBRAIC EQUATIONS ARE SOLVED BY STONE'S STRONGLY IMPLICIT SOLVER

Rockwell International
Rocketdyne Division

CFD 93 013-009/D1/AHH

KEMOD-1 MODULE DECK (CONT'D)

- **SUBROUTINE GRIDG**

  READS GRID NODE LOCATIONS PASSED FROM MAIN SOLVER AND FOR THE FIRST ITERATION CALCULATES GRID RELATED QUANTITIES (CELL AREAS AND VOLUME, NORMAL DISTANCES FROM SOLID BOUNDARIES AND INTERPOLATION FACTORS)

- **SUBROUTINE CALCE**

  ASSEMBLES THE COEFFICIENTS AND SOURCE TERMS FOR THE DISCRETIZED K AND ε TRANSPORT EQUATIONS IN THE FORM

$$A_p \phi_p = \sum_{i = E, W, N, S} A_i \phi_i + S\phi$$

  THE SUBROUTINE SOLVES THE DISCRETIZED EQUATIONS AFTER MODIFYING THE SOURCES AND BOUNDARY CONDITIONS FOR THE PARTICULAR PROBLEM

Rockwell International
Rocketdyne Division

CFD 93 013-010/D1/AHH

KEMOD-1 MODULE DECK (CONT'D)

- SUBROUTINE TWOLAY

  CALLED IF THE TWO-LAYER OR THE LOW-REYNOLDS NUMBER MODELS ARE USED. IT CALCULATES THE COEFFICIENTS NEEDED TO DESCRIBE THE ENERGY DISSIPATION AND EDDY VISCOSITIES CLOSE TO A WALL

- SUBROUTINE SOLSIP

  SOLVES THE SYSTEM OF ALGEBRAIC K AND ε EQUATIONS USING STONE'S STRONGLY IMPLICIT METHODS

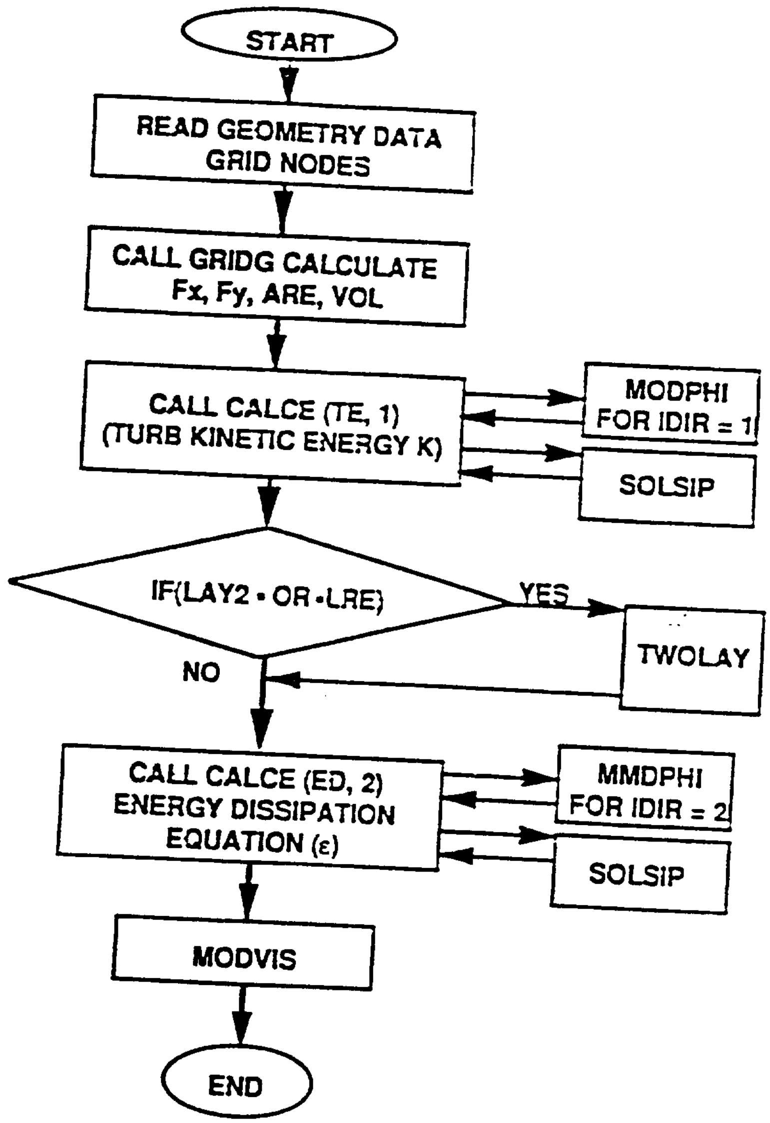
- SUBROUTINE USERM

  THIS SUBROUTINE HAS DIFFERENT ENTRY SECTIONS WHERE VARIABLES ARE UPDATED AND BOUNDARY CONDITIONS ARE SET

- SUBROUTINE MODIFY

  THIS IS THE ONLY SUBROUTINE THAT HAS TO BE CALLED FROM THE MOMENTUM EQUATION SOLVER OF THE MAIN ROUTINE. IT UPDATES THE FLUX SOURCE TERM OF THE DISCRETIZED MOMENTUM EQUATION DUE TO WALL SHEAR STRESSES

KEMOD-1 MODULE DECK (CONT'D)

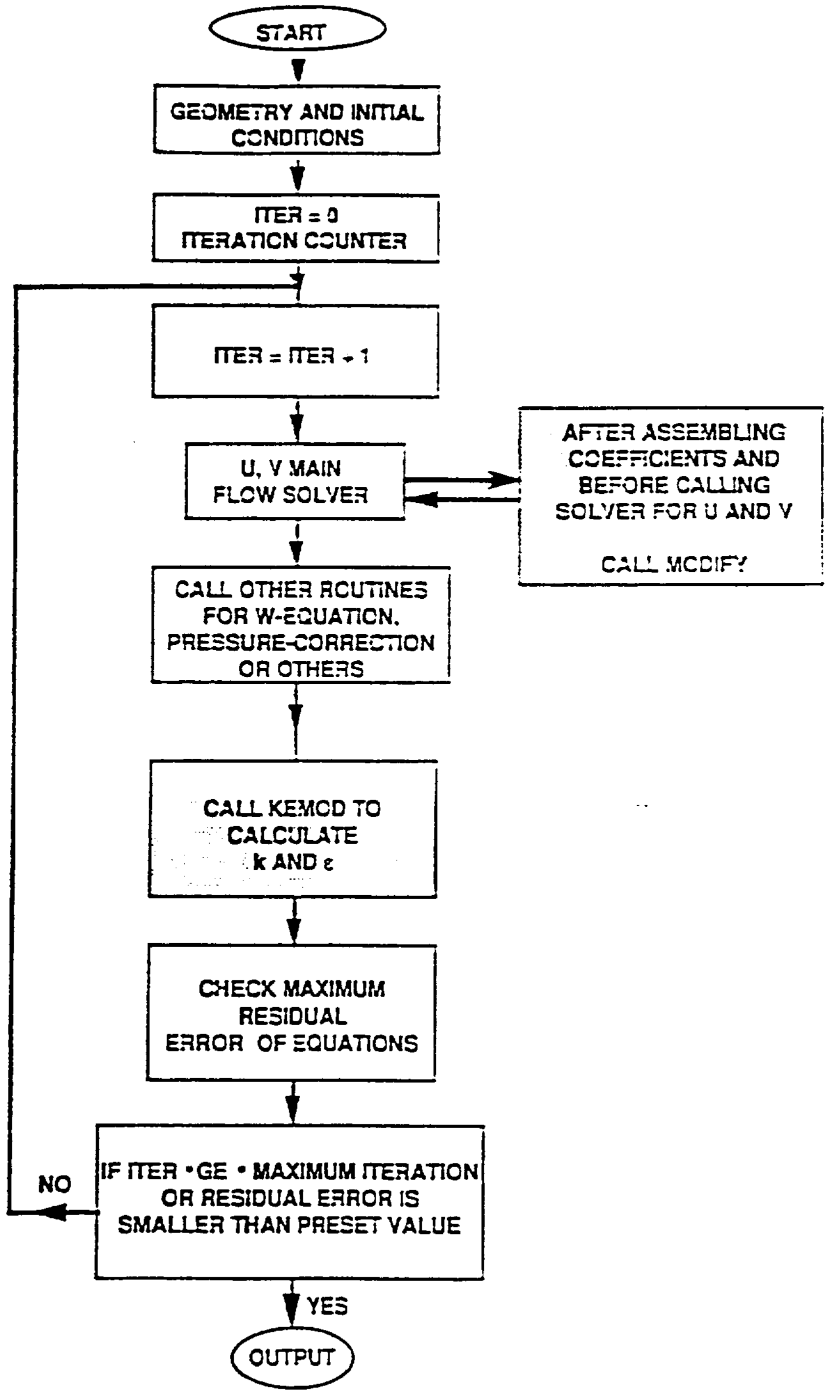


KEMOD FLOW CHART

Rockwell International
Rocketdyne Division

CFD 93 013-012/D1/AHH

KEMOD-1 MODULE DECK (CONT'D)



KEMOD-1 INTERFACE WITH A MAIN SOLVER

Rockwell International
Rocketdyne Division

CFD 93 013-013/D1/AHH

MULTI-SCALE k-ε MODEL

- DERVIED BY PARTITIONING THE TURBULENT ENERGY SPECTRUM INTO TWO SETS OF WAVE NUMBER REGIONS (PRODUCTION AND DISSIPATION RANGES) GIVING TWO EVOLUTION EQUATIONS FOR EACH REGION

- PARTITION LOCATION IS DETERMINED AS PART OF THE SOLUTION

- TURBULENT KINETIC ENERGY IN THE PRODUCTION RANGE OF THE SPECTRUM

$$\phi = k_p, \Gamma_{\phi x} = \Gamma_{\phi r} = \mu + \frac{\mu_t}{\sigma_{kp}}$$

$$S_{kp} = G - \rho \varepsilon_p$$

Rockwell International
Rocketdyne Division

CFD 93 013-014/D1/AHH

MULTI-SCALE k-ε MODEL (CONT'D)

- **ENERGY TRANSFER RATE IN THE PRODUCTION RANGE OF THE SPECTRUM**

$$\phi = \varepsilon_p, \quad \Gamma_{\phi x} = \Gamma_{\phi r} = \frac{\mu_t}{\sigma k_p}$$

$$S_{\varepsilon p} = \frac{1}{\rho} C_{p1} \frac{G^2}{K_p} + C_{p2} \frac{G \varepsilon_p}{k_p} - \rho\, C_{p3} \frac{\varepsilon_p^2}{k_p}$$

- **TURBULENT KINETIC ENERGY IN THE DISSIPATION RANGE OF THE SPECTRUM**

$$\phi = k_t, \quad \Gamma_{\phi x} = \Gamma_{\phi r} = \frac{\mu_t}{\sigma k_t}$$

$$S_{k_t} = \rho \varepsilon_p - \rho \varepsilon_t$$

MULTI-SCALE k-ε MODEL (CONT'D)

- **ENERGY DISSIPATION RATE IN THE DISSIPATION RANGE**

$$\phi = \varepsilon_t, \Gamma_{\phi x} = \Gamma_{\phi r} = \mu + \frac{\mu_t}{\sigma_{\varepsilon t}}$$

and

$$S_{\varepsilon t} = \rho C_{t1} \frac{\varepsilon_p^2}{k_t} + \rho C_{t2} \frac{\varepsilon_t \varepsilon_p}{k_t} - \rho C_{t3} \frac{\varepsilon_t^2}{k_t}$$

- **MODEL IS SIMILAR TO THAT USED BY KIM AND CHEN WITH CONSTANTS**

$$\sigma_{kp} = 0.75, \ \sigma_{\varepsilon p} = 1.15, \ \sigma_{kt} = 0.75, \ \sigma_{\varepsilon t} = 1.15$$

$$C_{p1} = 0.21, \ C_{p2} = 1.24, \ C_{p3} = 1.84, \ C_{t1} = 0.29$$

$$C_{t2} = 1.28, \ C_{t3} = 1.66 \text{ and } C_\mu = 0.09$$

Rockwell International
Rocketdyne Division

CFD 93 013-015A/D1/AHH

KEMOD-2 MODULE DECK

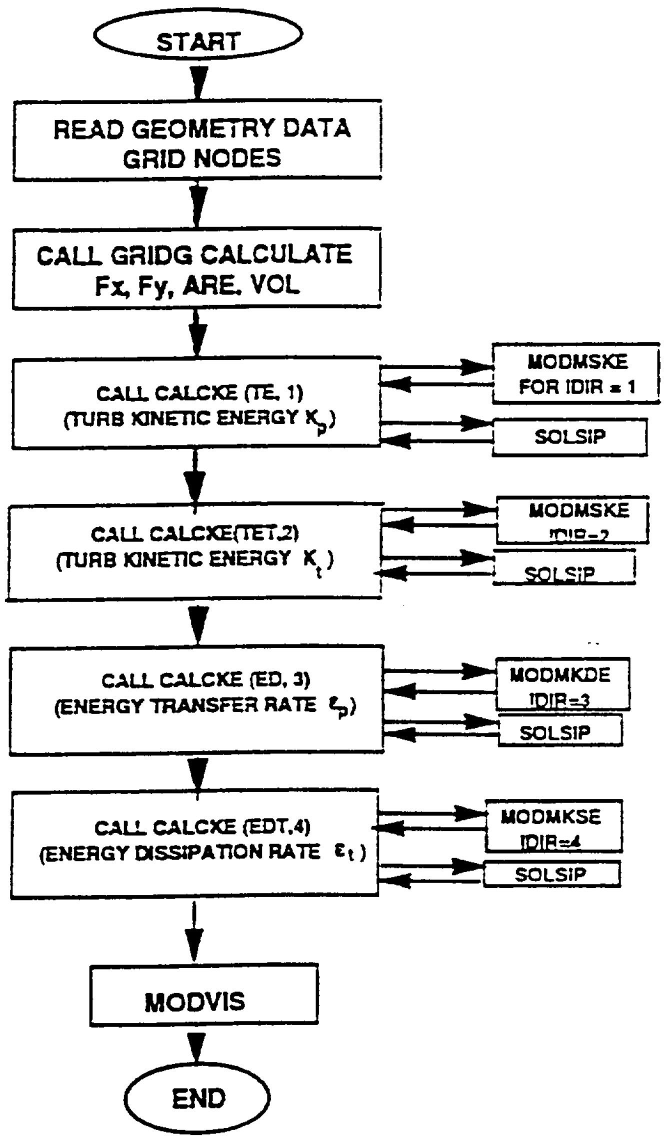
- KEMOD-2 IS A MULTI-TIME SCALE K-ε TURBULENCE MODULE DECK. IT CONSISTS OF TWO MAIN ROUTINES KEMOD AND MODIFY

- KEMOD IS CALLED WITHIN THE ITERATION LOOP OF THE MAIN FLOW SOLVER

- MEAN VELOCITIES AND OTHER VARIABLES ARE PASSED TO THE MODULE THROUGH ITS ARGUMENT LIST (EXPLAINED IN THE USER'S MANUAL

- THE MODULE IS STRUCTURED IN A SIMILAR WAY TO KEMOD-1 AND SUBROUTINE CALCE ASSEMBLES THE COEFFICIENTS AND SOURCE TERMS FOR THE DISCRETIZED K_p, ε_p, K_t, ε_t TRANSPORT EQUATIONS

KEMOD-2 MODULE DECK (CONT'D)

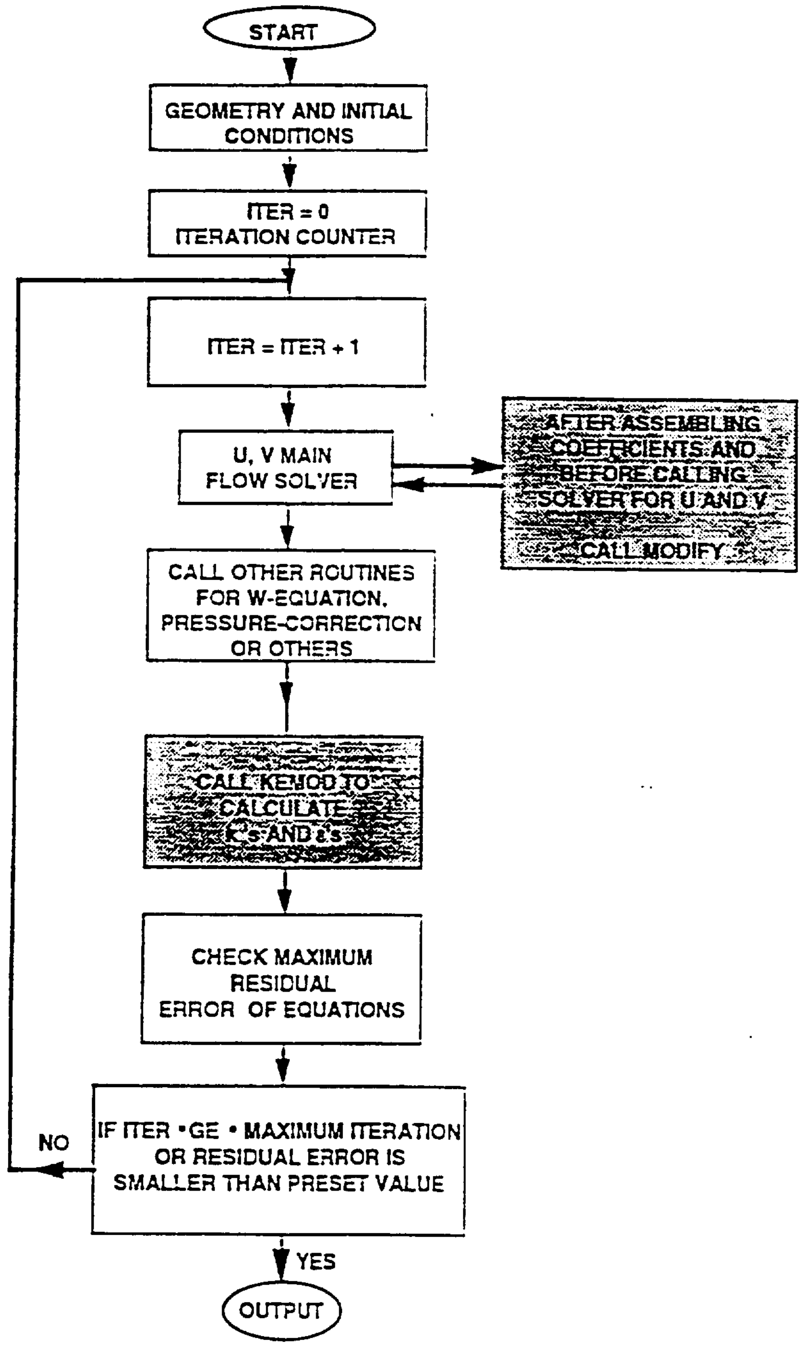


KEMOD-2 FLOW CHART

Rockwell International
Rocketdyne Division

CFD 93 013-017/D1/AHH

KEMOD-2 MODULE DECK (CONT'D)



KEMOD-2 INTERFACE WITH MAIN SOLVER

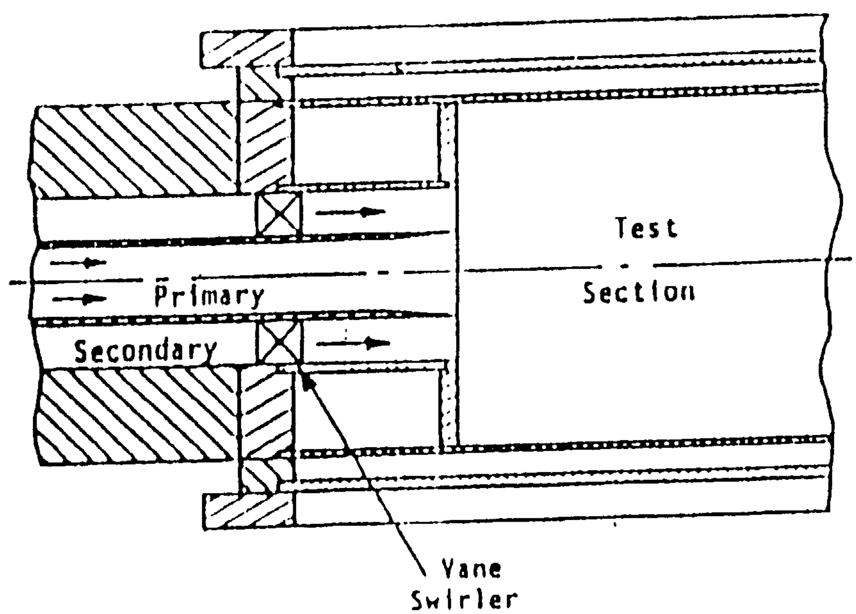
Rockwell International
Rocketdyne Division

CFD 93 013-018/D1/AHH

ROBACK AND JOHNSON – SWIRLING COAXIAL JETS DISCHARGING INTO AN EXPANDED DUCT

R. ROBACK AND B. JOHNSON, "MASS AND MOMENTUM TURBULENT TRANSPORT EXPERIMENT WITH CONFINED SWIRLING COAXIAL JETS," NASA CR-168252, 1983

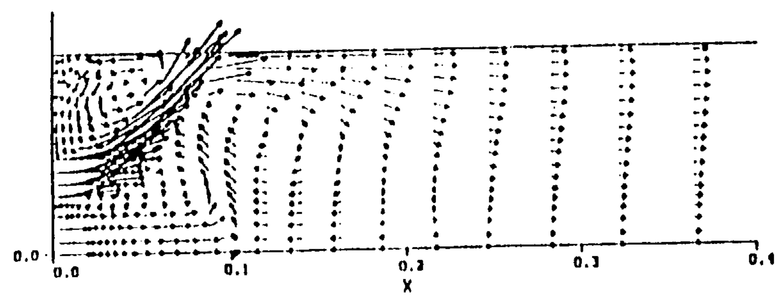
GEOMETRY



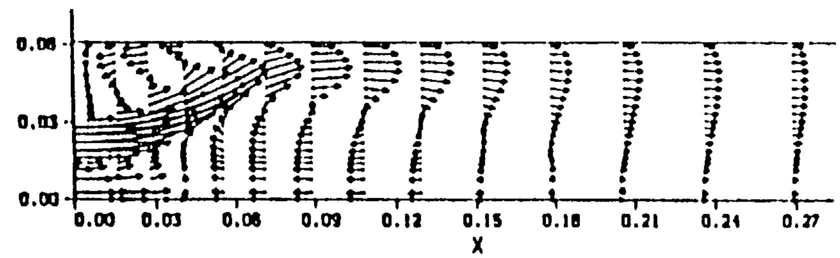
Rockwell International
Rocketdyne Division

CFD 92-032-0019/D3/MMS

ROBACK AND JOHNSON RESULTS
VELOCITY VECTORS



SINGLE-SCALE K-ε MODEL



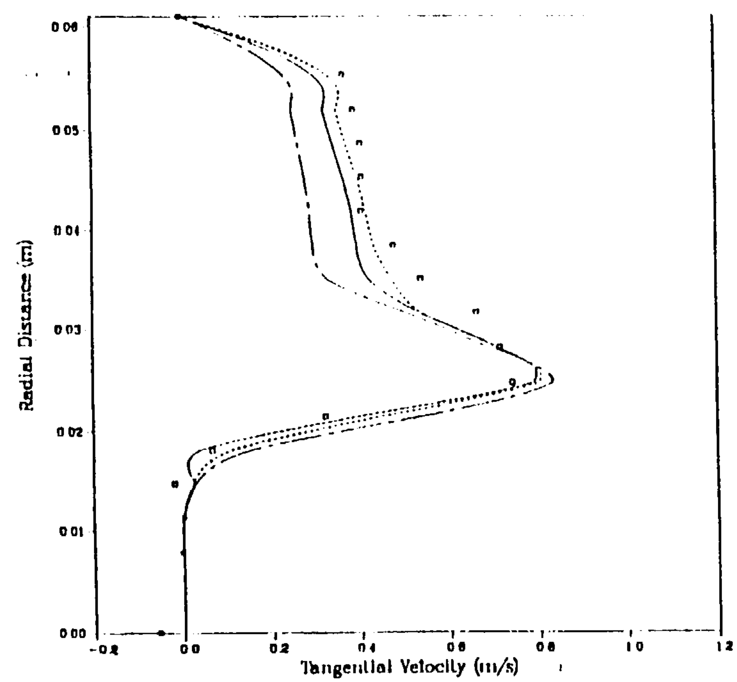
MULTI-SCALE K-ε MODEL

Rockwell International
Rocketdyne Division

CFD 92-032-020/D3/MMS

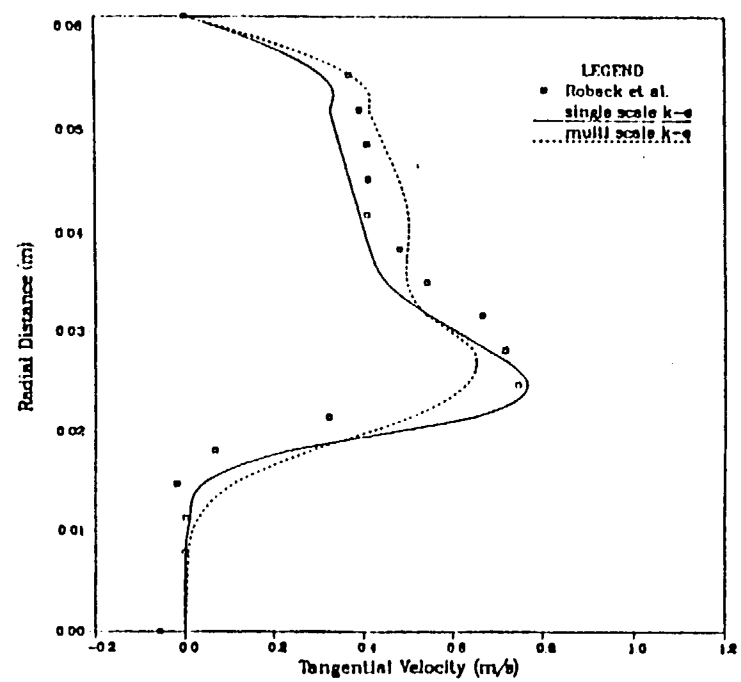
ROBACK AND JOHNSON RESULTS
TANGENTIAL VELOCITY PREDICTIONS AT X = 0.025 M

□ DATA
——— WALL FUNCTION
— — — LOW-REYNOLDS NO. MODEL
–·–·– 2-LAYER MODEL



SINGLE-SCALE K-ε

□ DATA
——— SINGLE-SCALE K-ε MODEL
— — — MULTI-TIME SCALE K-ε MODEL



Rockwell International
Rocketdyne Division

CFD 93 013-021/D1/AHH

SUMMARY

- **KEMOD-1 (2-D)**
  - SINGLE SCALE k-ε TURBULENCE MODULE COMPLETE
  - TESTED USING REACT AND USA CODES
- **KEMOD-2 (2-D)**
  - MULTISCALE k-ε TURBULENCE MODULE COMPLETE
  - TESTED USING REACT CODE
- **DEVELOPMENT OF MODULES FOR FULL AND ALGEBRAIC REYNOLDS STRESS MODELS IN PROGRESS**
- **WORK ON 3-D MODULES TO BEGIN AS SCHEDULED (FY '94)**

Rockwell International
Rocketdyne Division

CFD 93-013-029/D1/AHH

COMPUTATIONS OF CONFINED SWIRLING FLOWS WITH HIGH ORDER TURBULENCE MODELS IN A MODULAR FORM

A. Hadid[1], M. Sindir[1], C. Chen[2], and H. Wei[2]
[1] CFD Technology Center
Rocketdyne Division/Rockwell International
Mail Code IB39, 6633 Canoga Avenue
Canoga Park, CA 91303
[2] Department of Chemical Engineering
University of Alabama at Huntsville
Huntsville, AL 35899

Abstract

A finite-volume procedure is used to compare the performance of different high order turbulence models for confined swirling turbulent flows. Eddy-viscosity single and multi-scale k-ε turbulence models together with second-moment algebraic and Reynolds stress closure models are tested for a two-dimensional, axisymmetric swirling flow case. The ability of second-moment closure models to capture the interaction between swirl and the turbulent stress field is crucial to the predictive performance of the computational scheme.

To enhance the predictive capability of CFD tools for engineering applications, advanced turbulence models are coded as self-contained module decks that can be interfaced with a number of CFD solvers. Three of these modules, namely the single and the multi-scale models and the Algebraic stress model (ASM) have been successfully interfaced and tested with the code MAST of the University of Alabama at Huntsville in a relatively short time. These modules are independently tested and evaluated with the data of Roback and Johnson for swirling turbulent flow in a confined double concentric jets with a sudden expansion.

Modularization of a general purpose CFD code structure in terms of different aspects of physical models is necessary for computational efficiency. Further, individual modular routines are transportable and can be easily modified to include extra physical effects. This would allow many users using different CFD codes to concentrate their talents on developing and improving physical hypothesis for specific engineering problems.

Introduction

Computational Fluid Dynamics (CFD) has been used extensively for the last decade or so in analyzing complex flow phenomena for many industrial applications, such as, turbomachinery and combustion devices. Most flows of technological interest are turbulent and for many of them, relatively simple prediction methods are sufficient to produce results of engineering accuracy. For others, mainly in high technology applications, accurate predictions using high order turbulence models are required. Increases in available computational capabilities have permitted the development and testing of sophisticated models in the numerical simulation of turbulent flows. Direct numerical simulation, where all essential scales of the turbulent flow are resolved by solving the unsteady Navier-Stokes equations, are possible only at low to moderate Reynolds numbers. Turbulent flow analysis for engineering applications, therefore, can only be achieved by utilizing the time-averaged Navier-Stokes equations coupled with some level of modelling. The analysis of turbulent transport and modelling evolves from the Reynolds-averaged Navier-Stokes equations and auxiliary equations for velocity and length scales for eddy viscosity specifications towards a more sophisticated modeling startegy - one offering greater width of applicability, particularly in complex shear flows or where external force fields modify the turbulence structure.

One of the widely used models is the two-equation single-time-scale k-ε model(1). In this model transport equations for the turbulence energy (k) and the energy dissipation (ε) are solved to determine the turbulent eddy viscosity. An improvement to the single scale k-ε model is the multi-time-scale k-ε model where the energy spectrum of a turbulent flow is split into a production range and a dissipation range(2). Improved predictions using the multi-scale over the single-scale k-ε model have been demonstrated(3,4).

Other complicated single-scale models offering greater width of applicability, particularly in complex shear flows or where external force fields modify the turbulence structure are based on seconmoment closures. These take the exact equations for the transport of the Reynolds stresses ($\overline{u_i u_j}$) as their starting point and devise approximations for the unknown turbulent correlations appearing in them. In a three-dimensional flow, or even in an axisymmetric flow, all six components of the Reynolds stress tensor are nonzero. With a full second-moment closure model (RSM), therefore, differential transport equations need to be solved over the solution domain for each of these components. This represents an increase in the task of numerical solution compared with the situation where the k-ε eddy-viscosity model is adopted. An intermediate level of modeling has evolved[5,6] known as Algebraic second-moment closure (ASM), with the aim of retaining the greater physical realism of second-moment treatments while achieving computational times closer to that of an eddy-viscosity model. The simplification is achieved by approximating the convective and diffusive transport of the Reynolds stresses in terms of the corresponding transport of turbulent energy. This allows the transport equations for the stresses to be expressed as a set of algebraic formulae containing the turbulence energy and its rate of dissipation as unknowns. Second moment schemes have been extensively and successfully applied to a wide range of flows, as reviewed for example by Leschziner[7]. Few applications, however, have considered axisymmetric swirling flows[6,8] where the external forces due to swirl exert damping effects on the turbulent transport.

Progress in turbulence modelling have been paralleled by improvements in numerical techniques, essentailly, combining second moment closure with non-orthogonal, co-located grids using finite-volume methods. However, the implementation of RSM into non-orthogonal finite-volume codes poses difficulties: the co-located variable arrangement can cause decoupling of the mean velocity and Reynolds stress fields leading to oscillating solutions or even divergence. Using a special interpolation procedure in the context of Rhie[9], Obi and Peric[10] calculated the two-dimensional turbulent flow on a co-located grid arrangement usin the Reynolds stress turbulence model.

In the present paper, we pesent predictions of two dimensional/axisymmetric swirling flow using various models based on eddy-viscosity single and multi-scale k-ε and on second moment closure. These models are cast in a modular form enabling them to be used with a number of flow solvers based on the finite-volume and finite-difference methods. A discussion of the different models used and their assessment is presented. The modular structure of the different turbulence models will also be presented and discussed.

Theory and Model Equations

The turbulent flow considered is two-dimensional and steady which can be described by the Reynolds averaged continuity and momentum equations which may, respectively, be written as

$$\frac{\partial \rho U}{\partial x} + \frac{1}{r}\frac{\partial \rho r V}{\partial r} = 0 \qquad (1)$$

$$\frac{\partial \rho r U \Phi}{\partial x} + \frac{\partial \rho r V \Phi}{\partial r} = \frac{\partial}{\partial x}\left(r\mu \frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial r}\left(r\mu \frac{\partial \Phi}{\partial r}\right) + rS_{\Phi} \qquad (2)$$

Where Φ stands for any of the momentum components U, V, and rW and the corresponding sources S_{Φ} are

$$S_U = -\frac{\partial P}{\partial x} - \frac{\partial \rho \overline{u^2}}{\partial x} - \frac{1}{r}\frac{\partial \rho r \overline{uv}}{\partial r}$$

$$S_V = -\frac{\partial P}{\partial r} + \frac{rW^2}{r} - \frac{2\mu V}{r^2} - \frac{1}{r}\frac{\partial \rho r \overline{v^2}}{\partial r} - \frac{\partial \rho \overline{uv}}{\partial x} + \frac{r\overline{w^2}}{r}$$

$$S_r W = -\frac{2\mu}{r}\frac{\partial rW}{\partial r} - r\frac{\partial(\rho \overline{uw})}{\partial x} - r\frac{\partial(\rho \overline{vw})}{\partial r} - 2\rho\overline{vw}$$

where ρ, μ are the fluid density and visocosity respectively.

The appearance of the Reynolds stresses $\overline{u_i u_j}$ represents an unknown correlation and different turbulence models provide the means of relating these unknowns to known determinable quantities.

Single-Scale Eddy-Viscosity Turbulence Models

Here it is assumed that a single-time-scale (proportional to k/ε) can be used to describe the turbulent flow. Turbulence is simulated through transport equations for the turbulent kinetic energy (k), and its rate of dissipation (ε). The stress tensor is modelled using a gradient transport model of the form

$$-\overline{u_i u_j} = \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3} k\, \delta_{ij} \qquad (3)$$

The generalized form of the two-equation eddy-viscosity turbulence model can be written as
Kinetic Energy (k) equation:

$$C_k = D_k + P - \varepsilon \qquad (4)$$

where

$$C_k = \frac{\partial U_j k}{\partial x_j} \qquad \text{Convection of k}$$

$$D_k = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \qquad \text{Diffusion of k}$$

$$P = -\overline{u_i u_j}\frac{\partial U_i}{\partial x_j} = -\nu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j} \qquad \text{Production of k}$$

ν_t = eddy viscosity = $C_\mu \frac{k^2}{\varepsilon}$

Energy Dissipation (ε) equation:

$$C_\varepsilon = D_\varepsilon + \frac{\varepsilon}{k}(C_{\varepsilon 1}P - C_{\varepsilon 2}\varepsilon) \qquad (5)$$

where

$C_\varepsilon = \frac{\partial U_j \varepsilon}{\partial x_j}$ Convection of ε

$D_\varepsilon = \frac{\partial}{\partial x_j}[(\nu + \frac{\nu_t}{\sigma_\varepsilon})\frac{\partial \varepsilon}{\partial x_j}]$ Diffusion of ε

In the present study, the standard two-equation model was used with the wall function[1] and the two-layer model[11] to bridge the gap between the near-wall log-layer region and the fully turbulent region away from the wall. In the standard model the numerical values of the constants are C_μ=0.09, $C_{\varepsilon 1}$=1.44, $C_{\varepsilon 2}$=1.92, σ_k=1.0 and σ_ε=1.3. Details of the implementation of the wall function and the two-layer models can be found in Hadid and Sindir[12].

Multi-Time-Scale k-ε Turbulence Model

The Multi-time-scale turbulence model used here is based on the variable energy partitioning of the turbulent energy spectrum proposed by Kim and Chen[3]. In this model the turbulent kinetic energy spectrum is divided into two sets of wave number regions giving two evolution equations for each region. These equations represent the kinetic energy (k_p) and the energy dissipation (ε_p) in the production range of the spectum and the kinetic energy (k_t) and the energy dissipation (ε_t) in the dissipation range of the spectrum. This model allows the partition to move toward the high wave number region when production is high and toward the low wave number region when production vanishes.

The equations for the turbulent kinetic energy (k_p) and the energy transfer rate (ε_p) for the production range are

$$C_{kp} = D_{kp} + P - \varepsilon_p \qquad (6)$$

$$C_{\varepsilon p} = D_{\varepsilon p} + \frac{P}{\rho k_p}(\frac{1}{\rho}C_{p1}P + C_{p2}\varepsilon_p) - C_{p3}\frac{\varepsilon_p^2}{k_p} \qquad (7)$$

The equations for the turbulent kinetic energy (k_t) and the dissipation rate (ε_t) for the high wave number transfer region are

$$C_{kt} = D_{kt} + \varepsilon_p - \varepsilon_t \qquad (8)$$

$$C_{\varepsilon t} = D_{\varepsilon t} + \frac{\varepsilon_p}{k_t}(C_{t1}\varepsilon_p + C_{t2}\varepsilon_t) - C_{t3}\frac{\varepsilon_t^2}{k_t} \qquad (9)$$

where $C_{kp} = \rho\frac{\partial U_j k_p}{\partial x_j}$ and $C_{\varepsilon p} = \rho\frac{\partial U_j \varepsilon_p}{\partial x_j}$

$D_{kp} = \frac{\partial}{\partial x_j}[(\mu + \frac{\mu_t}{\sigma_{kp}})\frac{\partial k_p}{\partial x_j}]$ and

$D_{\varepsilon p} = \frac{\partial}{\partial x_j}[(\mu + \frac{\mu_t}{\sigma_{\varepsilon p}})\frac{\partial \varepsilon_p}{\partial x_j}]$

similarly for C_{kt}, $C_{\varepsilon t}$, D_{kt}, and $D_{\varepsilon t}$ equations and the model constants used are those of Kim and Chen[3].

The terms $(\frac{1}{\rho} C_{p1} \frac{P^2}{k_p})$ and $(\rho C_{t1} \frac{\varepsilon_p^2}{k_t})$ represent variable energy transfer functions. The former increases the energy transfer rate when production is high and the latter increases the dissipation rate when the energy transfer rate is high. The turbulent viscosity mt is given as $\mu_t = \rho C_\mu f k^2/\varepsilon_p = \rho C_\mu k^2/\varepsilon_t$, where $k=k_p+k_t$ is the total turbulent kinetic energy.

Second Moment Closure Models

The exact form of the Reynolds stress equations can be derived from the time-averaged form of the Navier-Stokes equations and can be written as:

$$\frac{D}{Dt}(\rho\overline{u_i u_j}) = P_{ij} + \Phi_{ij} + D_{ij} + \rho\varepsilon_{ij} \qquad (10)$$

where

$P_{ij} = -\rho\,(\overline{u_j u_k}\frac{\partial U_i}{\partial x_k} + \overline{u_i u_k}\frac{\partial U_j}{\partial x_k})$ Production

$D_{ij} = \frac{\partial}{\partial x_k}(-\rho\,\overline{u_i u_j u_k} - \delta_{ik}\,\overline{pu_i} - \delta_{jk}\,\overline{pu_j})$ Diffusion

$\Phi_{ij} = \overline{p\,(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})}$ Pressure-strain redistribution

$\varepsilon_{ij} = \frac{2}{3}\delta_{ij}\,\varepsilon$ Dissipation

Due to the introduction of correlations of higher orders, modelling of these terms is required to close the set of equations.

Algebraic Stress Model (ASM)

The ASM model used is based on the work of Rodi[5]. The idea is to simplify the stress equation (eq. 10) by approximating the convective and diffusve transport of the Reynolds stresses ($\overline{u_i u_j}$) in terms of the corresponding transport of turbulent energy. This simplification allows the transport equation of the stresses to be expressed as a set of algebraic formulae containing the turbulent energy and its rate of dissipation as unknowns. This set of algebraic equations can be written as;

$$\overline{u_i u_j} = \frac{k}{P-\varepsilon}\left[P_{ij} - \frac{2}{3}\delta_{ij}\varepsilon + \Phi_{ij} \right] \qquad (11)$$

The pressure-strain term Φ_{ij} is decomposed into a fluctuating part ($\Phi_{ij,1}$), a part due to the mean rate of strain ($\Phi_{ij,2}$), and a part due to reflected wall-influence ($\Phi_{ij,w}$), i.e., $\Phi_{ij} = \Phi_{ij,1} + \Phi_{ij,2} + \Phi_{ij,w}$

Rotta's return to isotropy concept is used to model the non-linear part ($\Phi_{ij,1}$) as

$$\Phi_{ij,1} = -C_1\frac{\varepsilon}{k}\left(\overline{u_i u_j} - \frac{2}{3}k\delta_{ij} \right)$$

$\Phi_{ij,2}$ is modelled using the isotopization of production concept as

$$\Phi_{ij,2} = -C_2\left(P_{ij} - \frac{2}{3}P\delta_{ij} \right)$$

The wall reflection term $\Phi_{ij,w}$ is modelled following Shir[13] and Gibson and Launder[14] as

$$\Phi_{ij,w} = \Phi_{ij,1w} + \Phi_{ij,2w}$$

where

$\Phi_{ij,1w}$ =

$$C_1'\rho\frac{\varepsilon}{k}\left(\overline{u_k u_m}\, n_k n_m \delta_{ij} - \frac{3}{2}\overline{u_k u_i}\, n_k n_j - \frac{3}{2}\overline{u_k u_j}\, n_k n_i \right) f \qquad (12)$$

$\Phi_{ij,2w}$ =

$$C_2'\left(\Phi_{km,2}\, n_k n_m \delta_{ij} - \frac{3}{2}\Phi_{ik,2}\, n_k n_j - \frac{3}{2}\Phi_{jk,2}\, n_k n_i\right) f \qquad (13)$$

where n_i is the wall-normal unit vector in the i-direction. The wall distance function (f) represents the ratio of turbulence length scale and the wall distance $f = \left(\frac{C_m^{0.75} k^{1.5}}{\kappa\,\varepsilon}\right)\frac{1}{\Delta n}$ where Δn is the wall-normal distance. The above wall-correction terms are written in a tensorialy invariant form and their effect is to transfer energy from the wall-normal normal stress component to the tangential stresses i.e it is redistributive.

For axisymmetric swirling flows the set of algebraic stress equations can be written in a general matrix form as $\underline{A}\ \underline{T} = \underline{B}$ where

$\underline{A}$ =

$$\begin{matrix}
\frac{3\varepsilon}{2\lambda k}+2\frac{\partial U}{\partial x} & \frac{\partial V}{\partial y} & -\frac{V}{r} & 2\frac{\partial U}{\partial y}-\frac{\partial V}{\partial x} & -\frac{\partial W}{\partial y}+\frac{W}{r} & -\frac{\partial W}{\partial x} \\
-\frac{\partial U}{\partial x} & \frac{3\varepsilon}{2\lambda k}+2\frac{\partial V}{\partial y} & -\frac{V}{r} & 2\frac{\partial V}{\partial x}-\frac{\partial U}{\partial y} & -\left(\frac{\partial W}{\partial y}+2\frac{W}{r}\right) & -\frac{\partial W}{\partial x} \\
-\frac{\partial U}{\partial x} & \frac{\partial V}{\partial y} & \frac{3\varepsilon}{2\lambda k}+2\frac{V}{r} & -\left(\frac{\partial U}{\partial y}+\frac{\partial V}{\partial x}\right) & 2\frac{\partial W}{\partial y}+\frac{W}{r} & 2\frac{\partial W}{\partial x} \\
\frac{\partial V}{\partial x} & \frac{\partial U}{\partial y} & 0 & \frac{\varepsilon}{\lambda k}+\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y} & 0 & -\frac{W}{r} \\
0 & \frac{\partial W}{\partial y} & -\frac{W}{r} & \frac{\partial W}{\partial x} & \frac{\varepsilon}{\lambda k}+\frac{\partial V}{\partial y}+\frac{V}{r} & \frac{\partial V}{\partial x} \\
\frac{\partial W}{\partial x} & 0 & 0 & \frac{\partial W}{\partial y} & \frac{\partial U}{\partial y} & \frac{\varepsilon}{\lambda k}+\frac{\partial U}{\partial x}+\frac{V}{r}
\end{matrix}$$

$$\underline{T} = \left[\rho\overline{uu}, \rho\overline{vv}, \rho\overline{ww}, \rho\overline{uv}, \rho\overline{vw}, \rho\overline{uw} \right]^T$$

$$\underline{B} = \begin{matrix}
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}(\Phi_{11,1w} + \Phi_{11,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}(\Phi_{22,1w} + \Phi_{22,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)}(\Phi_{33,1w} + \Phi_{33,2w} \\
\frac{3}{2(1-C_2)}(\Phi_{12,1w} + \Phi_{12,2w}) \\
\frac{3}{2(1-C_2)}(\Phi_{23,1w} + \Phi_{23,2w}) \\
\frac{3}{2(1-C_2)}(\Phi_{13,1w} + \Phi_{13,2w})
\end{matrix}$$

and $\lambda = \dfrac{1-C_2}{C_1 - 1 + \frac{P}{\rho\varepsilon}}$

Reynolds Stress Model (RSM)

In the RSM model the full transport equation for the Reynolds stresses (eq.10) are solved for each stress component ($\overline{u_i u_j}$) after modelling the diffusion and the pressure strain terms similar to Launder et. al[15]. The diffusion term is modelled as

$$D_{ij} = -\frac{\partial}{\partial x_k}\left[\rho C_k\, \overline{u_k u_l}\, \frac{k}{\varepsilon}\frac{\partial \overline{u_k u_l}}{\partial x_k}\right]$$

The pressure-strain redistribution term Φ_{ij} is modelled in a similar way to that used in the ASM model discussed earlier. Special consideration is given to the problem of mean velocity-Reynolds stress decoupling which appear when using a collocated grid arrangement which is a source of numerical instability. This is done by invoking a special interpolation procedure for the cell-face stresses in the context of Rhie[9]. This practice results in the addition of normal stresses to the pressure term where the cell-face velocity is sensitized to the pressure differences as well as to normal stress diffrences at the nodes surrounding the face.

Turbulence Model Decks (Modules)

As the state-of-the-art of computers has advanced, so has the range, size and complexity of flow models being applied. Users have become more sophisticated and there is a constant demand for improvement. CFD codes have adapted to this demand and many general-purpose computer codes have been developed and used. As general-purpose codes become larger, their code structure becomes sophisticated. In general codes can be divided into three main areas, they include; 1) Numerical algorithms (which can be subdivided into discretization methods and solution techniques). 2) Methods of dealing with complex geometries. 3) Physical models (which include turbulence models, porosity, combustion kinetics, two-phase flow...). It seems, therefore, that the practicing engineer must have the knowledge of all these elements of the CFD program in order to successfully utilize this code. To obtain the maximum benefits from these general-purpose CFD codes, modularization of the code structure may be necessary. That is developing individual modular routines for the solver and for different physical models for example. If such modules are successful it would allow users to concentrate their talents on developing and improving physical hypothesis such as turbulent models for example that can easily be tested using such modules.

In the present work, turbulent modules are being developed to meet this need. Figure 1, shows a flowchart of a turbulence module interfaced with a typical main flow solver. The module is called by the flow solver passing to it the mean flow velocities, mass fluxes at cell faces and grid information among others. The turbulence differentail equations are discretized and the matrix coefficients are setup and solved using Stones strongly implicit method[16]. In the ASM module, the set of algebraic stress equations are solved simultanously using Gauss-Seidel method at each step or iteration. In the eddy-viscosity models the values of k, ε, and eddy viscosity (μ_t) are passed to the main flow solver, while, in the second moment closure models the Reynolds stresses $\overline{u_i u_j}$ are passed to the main solver. The solver then calls subroutine MODIFY of the module where the momentum sources are modified to account for the near-wall shear stresses in the eddy-viscosity models or to calculate Reynolds stress gradients in the second moment models.

These modules are structured to be self-contained and transportable to a number of general purpose CFD solvers to maximize computational efficiency. They have been tested independently at the University of Alabama at Huntsville using the MAST code.

Results

The various turbulence models are analyzed by comparing model predictions with the experimental data of Roback and Johnson[17] for swirling flow in confined double concentric jets with a sudden expansion.

Figure 2, shows the decay of the mean axial centerline velocity using both the single and multi-scale k-ε models. Figure 2a, shows the comparison using the wall-function near wall approach and Figure 2b shows the results using the two-layer near wall model. The single-scale k-ε model seems to underpredict the extent of the central recirculation zone as compared with the multi-scale k-ε model. Moreover, improved comparisons with the data are obtained using the two-layer near wall model. Figure 3, shows the radial profile of the mean axial velocity at two distances downstream of the jet exit. Again, the two-layer model predicts better comparisons with tha data than the wall function approach. The radial profiles of the mean tangential velocity are shown in Figure 4.

Figure 5, shows the radial profile of the mean axial velocity at three axial locations using the algebraic stress model (ASM) with wall function and two-layer near wall models. The radial profiles of the rms axial turbulent intensity are shown in Figure 6. Streamline contours are shown in Figure 7 using the single-scale k-ε and the ASM models with the two-layer near wall approach. The extent of the central vortex is better predicted using the ASM model. Preliminary results were also obtained using the full Reynolds stress model (RSM). Comparisons with the backward facing step data of Driver and Seegmiller[18] shows improved predictions over the single scale k-ε model as shown in Figure 8 where the radial profiles of the axial normal stress and shear stress are plotted at four step heights downstream. Further testing of the RSM model for swirling flows are planned.

Conclusions

Different turbulence models for industrial applications have been formatted in a modular form and successfully interfaced and tested independently using two different main flow solvers. The turbulence models include the single and multi-sclae k-ε models both with wall functions and two-layer near wall models. Second moment models that include the algebraic (ASM) and full Reynolds stress model (RSM) have been tested. It was shown that the two-layer near wall model improves predictions as compared to the wall function approach. Convergence of the stiff ASM model equations was obtained by solving the 6x6 stress equations (for axisymmetric/swirling flows) at each iteration. The wall-reflection terms in the pressure-strain model showed little or no improvements in the ASM model predictions. Elaborate pressure-strain models that require no wall-damping are needed e.g. Speziale et.al[19]. The full Reynolds stress model (RSM) promises to be the next model to be used for engineering applications.

References

(1) Launder B., and Spalding D., Comp. Meth. in Appl. Mech. & Eng. vol. 3, 1974, pp. 269-289
(2) Hanjalic K., Launder B., and Schiestel R., Turb. Shear Flows, eds. Bradbury L. et. al, vol.2, Springer-Verlag, N.Y. 1980, pp. 36-49
(3) Kim S.-W and Chen C.-P, Num. Heat Transfer, pt B, vol. 16, no.2, 1989, pp.193-211
(4) Kim S.-W, Num. Heat Transfer, pt B., vol. 17, 1990, pp.101-122
(5) Rodi W., Z. Ang. Math. und Mech., vol 56, 1976, pp.219-
(6) Fu S., Huang P., Launder B., and Leschziner M., ASME J. Fluid Eng., vol.110, 1988, pp.216-221
(7) Leschziner M., Proc. of Int. Form on Math. Modeling of Processes in Energy Systems, Int. Center for Heat and Mass Transfer, Sarajevo, 1989
(8) Hogg S., and Leschziner M., AIAA J., vol.27, no.1, 1989, pp.57-63
(9) Rhie C., Univ. of Ill., Urbana-Champaign, Ph.D. Thesis, 1981
(10) Obi S., and Peri'c M., AIAA J., vol.29, no.4, 1991, pp.585-590
(11) Chen H., and Patel V., AIAA J., vol. 26, no. 6, 1988, pp.641-648
(12) Hadid H., and Sindir M., NASA CP-3174, (1992)
(13) Shir C., J. Atmos. Sci., vol.30, 1973, pp.1327-
(14) Gibson M., and Launder B., J. Fluid Mech., vol.86, pt.3, 1978, pp.471-
(15) Launder B., Reece G., and Rodi W., J. Fluid Mech., vol.68, 1975, pp.537-566
(16) Stone H., SIAM J. num. anal. vol.5, 1968, pp.530-
(17) Roback R., and Johnson B., NASA-CR 168252 (1983)
(18) Driver D., and Seegmiller H., AIAA Paper 82-1029, (1982)
(19) Speziale C, Sarkar S., and Gatski T., J. Fluid Mech., vol.227, 1991, pp.245-272

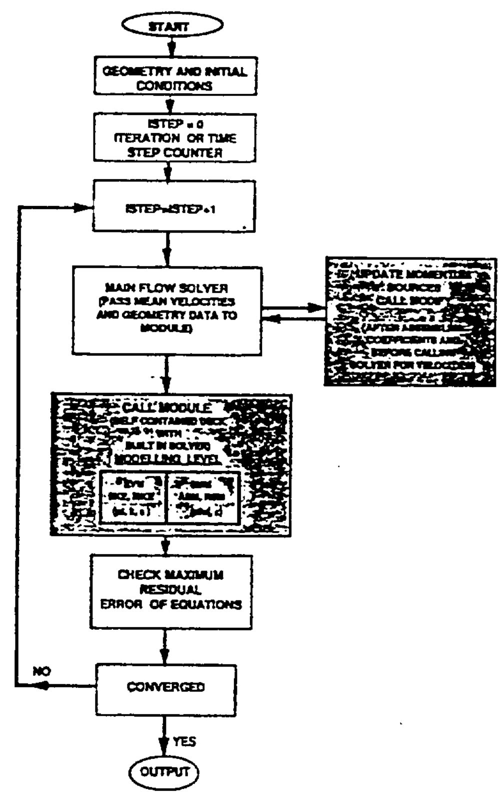


Figure 1. Typical main flow solver interfaced with a turbulence module

CONFINED SWIRLING JET FLOW (ROBACK & JOHNSON)
Decay of mean axial centerline velocity

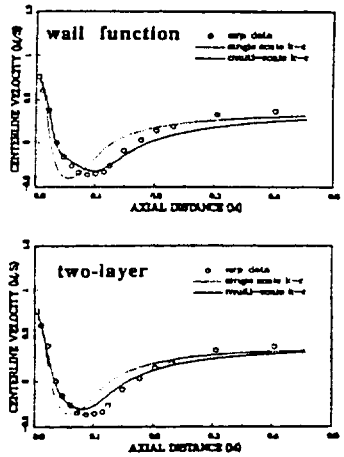


Figure 2. Mean axial centerline velocity
----- single-scale k-ε model
___ multi-scale k-ε model

CONFINED SWIRLING JET FLOW (ROBACK & JOHNSON)
— STANDARD k-ε, ___ MULTI-SCALE k-ε

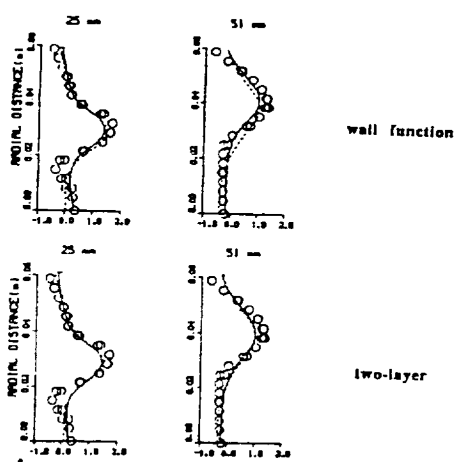


Figure 3. Radial profile of mean axial velocity

CONFINED SWIRLING JET FLOW (ROBACK & JOHNSON)
— STANDARD k-ε, ___ MULTI-SCALE k-ε

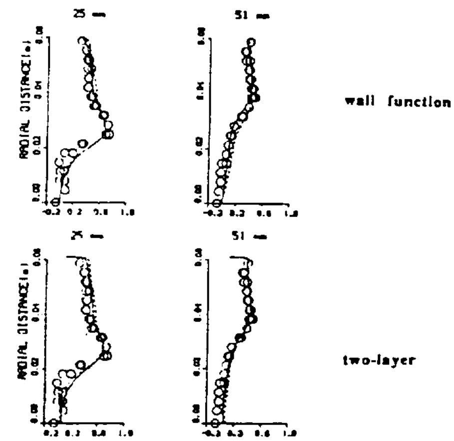


Figure 4. Radial profiles of mean tangential velocity

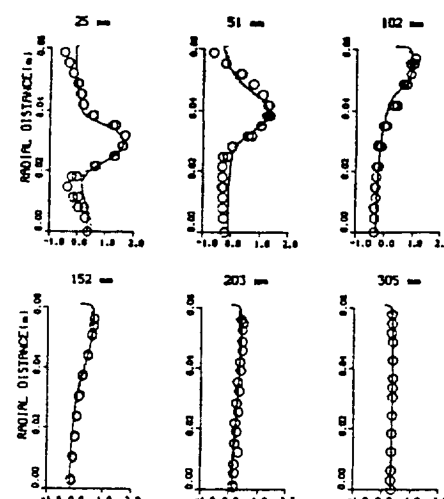


Figure 5. Radial profiles of mean axial velocity
ASM turbulence model
---- wall function model
___ two-layer model

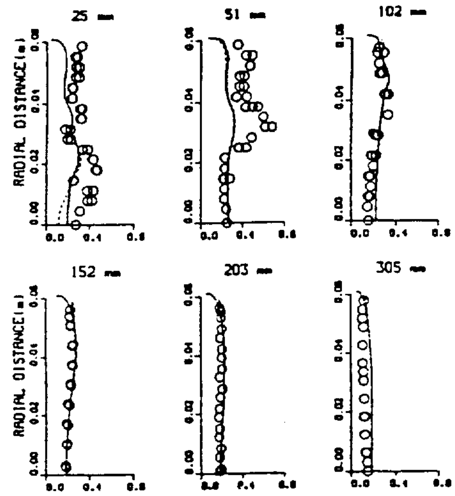
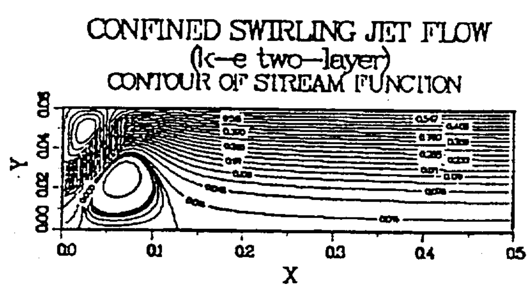


Figure 6. Radial profiles of the axial turbulent intensity
ASM turbulence model
--- wall function model
___ two-layer model



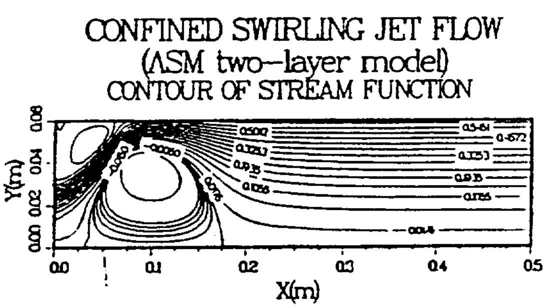


Figure 7. Sreamline contours

(a) axial turbulent intensity $(\overline{u^2})$

(b) shear stress $(\overline{uv})$

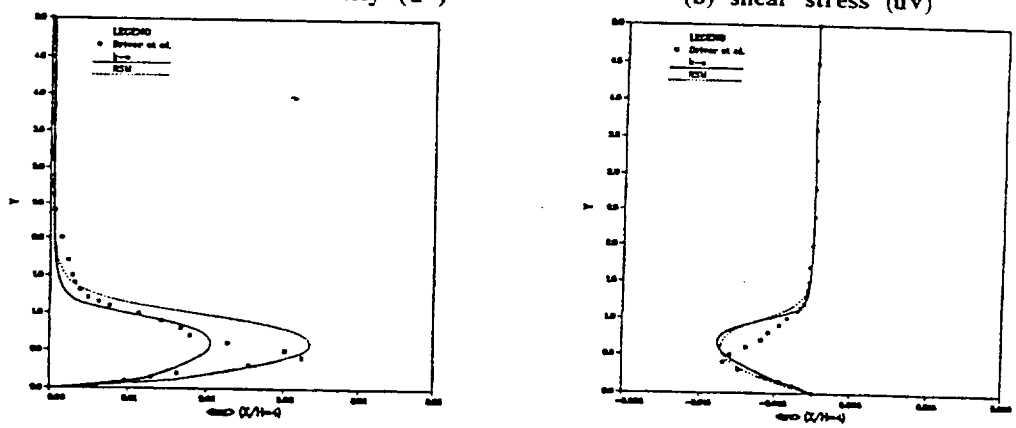


Figure 8. Backward facing step (Driver & Seegmiller)
—— single-scale k-ε model
--- RSM model

Computational Fluid Dynamics JOURNAL vol.1 no.2 July 1992 (207 — 214)

A NUMERICAL STUDY OF TWO-DIMENSIONAL VORTEX SHEDDING FROM RECTANGULAR CYLINDERS

A. H. HADID† M. M. SINDIR† R. I. ISSA‡

Abstract

An efficient time-marching, noniterative calculation method is used to analyze time-dependent flows around rectangular cylinders. The turbulent flow in the wake region of a square section cylinder is analyzed using an anisotropic k-ε model. Initiation and subsequent development of the vortex shedding phenomenon is naturally captured once a perturbation is introduced in the flow. Transient calculations using standard eddy-viscosity and anisotropic k-ε models, averaged over an integral number of cycles to get the fluctuating energy (organized and turbulent), are compared with experimental data. It is shown that the anisotropic k-ε model resolves the anisotropy of the Reynolds stresses and gives mean energy distribution closer to the experiment than the standard k-ε model.

1. INTRODUCTION

Vortex shedding is a periodic unsteady flow phenomenon that occurs frequently behind bluff bodies and is therefore of great practical importance. Many attempts to calculate the two-dimensional (2-D) vortex shedding motion past square and circular cylinders by solving the unsteady Navier-Stokes equations were successful at low Reynolds numbers where the flow is laminar and the fluctuations are periodic, e.g., [1] and [2]. At higher Reynolds numbers which are more relevant in practice, turbulent fluctuations are superimposed on the periodic unsteady motion. The problem then concerns the decomposition of the flow into organized motion that is resolved in the calculation and a remaining turbulent motion to be represented by a turbulence model. Previous analysis of vortex shedding calculations at high Reynolds numbers have not been successful due to the inadequacy of the standard k-ε model and the lack of affordable higher order models that take into account the anisotropy of the turbulent intensities

Franke *et al.* [3] analyzed the unsteady turbulent flow for a square cylinder using the standard k-ε model. They showed that the model tends to damp the periodic shedding motion underpredicting the Strouhal number. They also analyzed the detailed experimental results of Cantwell and Coles [4] for vortex shedding in the 2-D wake behind a circular cylinder. They additionally point out the need for improved models that account for the history and transport effects of the individual stresses. MacInnes *et al.* [5] used the standard k-ε model to simulate the periodically forced turbulent mixing layer investigated experimentally by Weisbrot and Wygnanski [6]. They managed to capture the main features of the mixing layer development where there is a clear distinction between the organized and the random turbulent motion.

Presented at the 4th International Symposium on Computational Fluid Dynamics - Davis on September 9 - 12, 1991. Received on March 21, 1992. Received in final form on May 12, 1992.

† CFD Technology Center, Rockwell Int./Rocketdyne Div., 6633 Canoga Ave., Canoga Pk., CA 91303, USA

‡ Dept. Mineral Resources Engineering, Imperial College of Science, Technology and Medicine, London, SW7, 2BP, England

Majumdar and Rodi [7] have shown that the separated turbulent flow past circular cylinders cannot be predicted realistically without a time-accurate numerical procedure to account for the periodic shedding of vortices.

Experimental investigations are needed to judge the different numerical and turbulent schemes. Durão *et al.* [8] conducted an experimental study of transient turbulent flow behind a square cylinder. They used spectral analysis and digital filtering of the LDV data in order to separate and quantify the turbulent and periodic, nonturbulent motions. They show for example that in the zone of highest velocity fluctuations the energy associated with the turbulent fluctuations is about 40 % of the total energy. Therefore, for a successful simulation of transient turbulent flows, a reliable time-accurate numerical procedure and a good turbulence model are needed.

The purpose of the present paper is to model turbulent vortex shedding flows using an efficient time-accurate numerical procedure based on the PISO [9] methodology. Calculations of the turbulent vortex shedding are performed using the two-equation k-ε model with isotropic eddy-viscosity and with a modified two-equation model using an anisotropic eddy-viscosity. In the anisotropic model, nonlinear corrections are added to improve the eddy-viscosity representation of the Reynolds stresses as developed by Yoshizawa [10] with the aid of a two-scale direct interaction approximation. A similar anisotropic eddy-viscosity model was also developed by Speziale [11]. The adequacy of the models to simulate transient turbulent flows is assessed with the aid of the experimental results of Durão *et al.* [8] for vortex shedding in the 2-D wake behind a square cylinder at $Re = 14,000$.

2. MODEL EQUATIONS

The basic equations of motion in transient periodic flows can be written after separating the flow into an organized (phase averaged) component

$$U_i(x_i, t) = \frac{1}{N} \sum_{n=0}^{N} u_i(x_i,\, t + nT) \tag{1}$$

where $U_i(x_i, t)$ is the resolvable portion of the instantaneous velocity u_i, and T is the period of the oscillation, and a random turbulent component $u'_i(x_i, t)$. The instantaneous velocity $u_i(x_i, t)$ is then given by

$$u_i = U_i + u'_i = \overline{u_i} + \widetilde{u_i} + u'_i \tag{2}$$

where $\overline{u_i}$ is the time-mean component of the velocity, and $\widetilde{u_i}$ is the periodic fluctuating component. Assuming an incompressible flow, the momentum equations can be written after applying phase averaging as;

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (\nu \frac{\partial U_i}{\partial x_j} + R_{ij}) \tag{3}$$

where $R_{ij} = - < u'_i u'_j >$ is the phase-averaged Reynolds stress tensor and ν is the kinematic viscosity.

Standard Isotropic k-ε Model

In the standard isotropic k-ε model [12], R_{ij} is approximated by using the eddy-viscosity ν_t as;

$$R_{ij} = -\frac{2}{3} k \delta_{ij} + \nu_t (\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}) \tag{4}$$

where k is the phase-averaged turbulent kinetic energy and $\nu_t = C_\mu(k^2/\varepsilon)$, ε is the phase-averaged energy dissipation rate and C_μ is a model constant. The spatial and temporal distribution of k and ε are determined from differential transport equations of these quantities

$$\frac{\partial k}{\partial t} + U_i \frac{\partial k}{\partial x_i} = \frac{\partial}{\partial x_i}\left[(\nu + \frac{\nu_t}{\sigma_k})\frac{\partial k}{\partial x_i}\right] + G - \varepsilon \tag{5}$$

$$\frac{\partial \varepsilon}{\partial t} + U_i \frac{\partial \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i}\left[(\nu + \frac{\nu_t}{\sigma\varepsilon})\frac{\partial \varepsilon}{\partial x_i}\right] + \frac{\varepsilon}{k}(C_1 G - C_2\varepsilon) \tag{6}$$

where $G = R_{ij}\dfrac{\partial U_i}{\partial x_j}$ is the turbulent generation term. The constants C_μ, C_1, C_2, σ_k, and σ_ε have values of 0.09, 1.44, 1.92, 1.0, and 1.3, respectively.

Anisotropic k-ε Model

In the anisotropic model the Reynolds stresses can be expressed as;

$$R_{ij} = -\frac{2}{3}k\delta_{ij} + \nu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) + \frac{1}{3}\left(\sum_{m=1}^{3}\tau_m Sm_{kk}\right)\delta_{ij} - \sum_{m=1}^{3}\tau_m Sm_{ij} \tag{7}$$

$$\tau_m = C\tau_m\frac{k^3}{\varepsilon^2} \tag{8}$$

$$S1_{ij} = \frac{\partial U_i}{\partial x_k}\frac{\partial U_j}{\partial x_k} \tag{9}$$

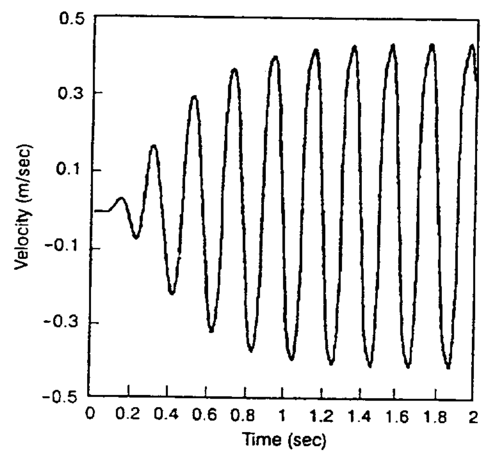
$$S2_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_k}\frac{\partial U_k}{\partial x_j} + \frac{\partial U_j}{\partial x_k}\frac{\partial U_k}{\partial x_i}\right) \tag{10}$$

$$S3_{ij} = \frac{\partial U_k}{\partial x_i}\frac{\partial U_k}{\partial x_j} \tag{11}$$

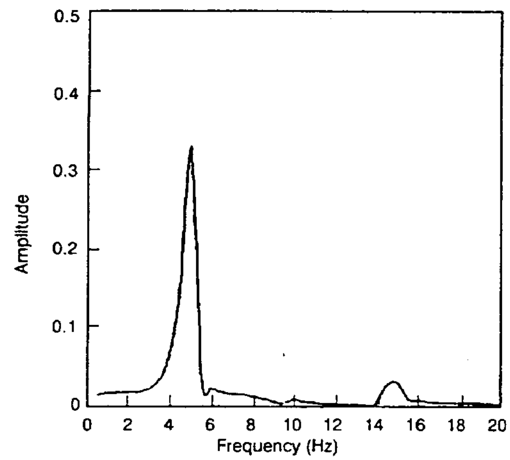
and $C\tau_m$ (m = 1, 2, 3) are model constants. The first two terms on the right hand side of (7) give the familiar isotropic eddy-viscosity representation, while the third and fourth terms express the anisotropy of R_{ij}. These additional nonlinear quadratic terms of the mean velocity gradients seem to be a simple way to resolve the individual normal stresses with the k-ε model. The anisotropy is reflected especially in the k-ε equation where both the diffusion and production terms are quadratic forms of the mean velocity gradients and turbulent kinetic energy gradients.

The anisotropic eddy-viscosity model has been successfully used by Nisizima and Yoshizawa [13] and Myong and Kasagi [14] for fully developed turbulent channel flows. In their calculations only $C\tau_1$ and $C\tau_2$ were optimized to reproduce the anisotropy of the turbulent intensities since $C\tau_3$ does not appear in their equations. In the present study the flow is shear dominated with little departure from isotropy. Therefore, the model constants $C\tau_1$, $C\tau_2$, and $C\tau_3$ were optimized to 0.01, 0.01, and 0.001, respectively, to satisfy the realizability constraint. (Note: zero constants reduce to the isotropic eddy-viscosity model.)

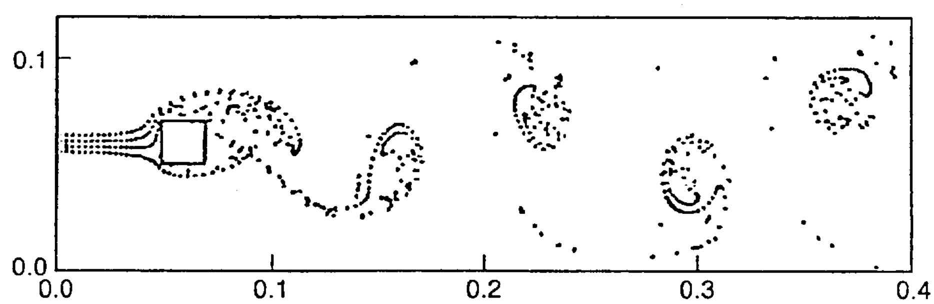
Applications of the k-ε isotropic and anisotropic eddy-viscosity models were made using wall functions to bridge the viscosity affected near the obstacle wall region. It is assumed that inadequacies in near-wall modelling play a minor role to the inaccuracy of normal Reynolds



(a) Normal velocity fluctuation along the centerline



(b) Power spectrum of the normal velocity fluctuation



(c) Streak line plot

Fig.1: Flow characteristics of the wake for $Re = 14,000$

stress differences arising from use of an isotropic eddy viscosity. Improvements can be made by integrating all the way to the wall [15] or by using the two-layer model of Chen and Patel [16].

3. NUMERICAL METHOD

The PISO methodology [9], in conjunction with a finite-volume technique, is used to solve the implicitly discretized, time-dependent flow equations. The method is essentially noniterative, where the solution process is split into a series of steps whereby operations on pressure are decoupled from those on velocity at each time-step. The avoidance of iterations substantially reduces the computational effort compared with that required by iterative methods. This is possible since the splitting error of PISO is negligibly small at the level of time-step required to eliminate the temporal truncation error. A backward temporal difference scheme is used, while the convective terms are discretized using a second-order upwind difference scheme. The method can also be used for steady-state flows, e.g., Hadid *et al.* [17].

Calculations are performed for the turbulent flow around a square cylinder (step height, $H = 20\,\text{mm}$) in a domain extending about 16 H downstream and 2.5 H upstream of the obstacle.

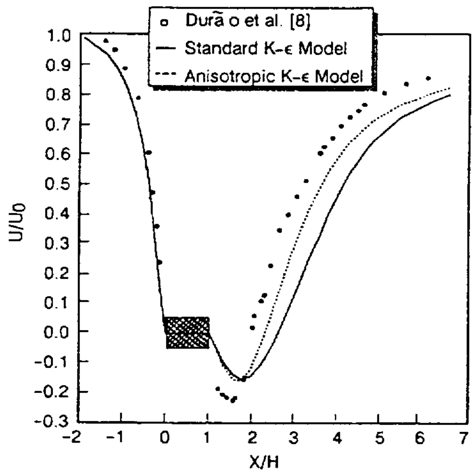


Fig.2: Centerline distribution of mean axial velocity

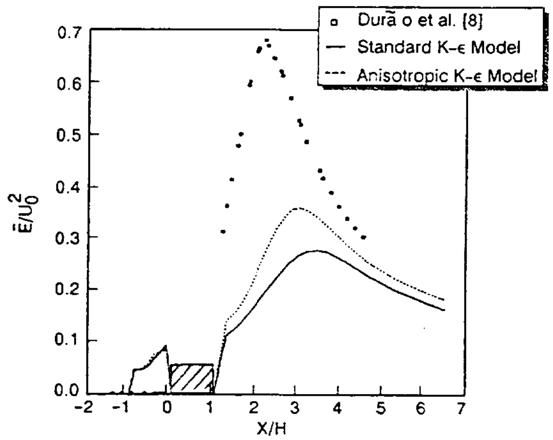


Fig.3: Time-mean kinetic energy of the velocity fluctuations

The calculations captured the vortex shedding phenomenon after perturbing the flow at the inlet. A reference velocity of 0.68 m/s and turbulence intensity of 6 % (i.e., $k =< u'^2 >= 3.6\times10^{-3}\,\mathrm{m^2/s^2}$) were used as the inlet conditions. The length scale L of turbulence at the inlet was not measured in the experiment but an order of $L \sim 0.1$ mm was assumed from which the energy dissipation rate $\varepsilon = k^{3/2}/L$ was estimated. It is expected that the calculated results are not sensitive to the precise value of ε used at the inlet. The upper and lower boundaries are treated as symmetry planes, at the exit, a zero-gradient outflow boundary condition is applied to each variable. The computational domain is resolved by 75×40 grid cells with clustering at the obstacle walls. An optimized time step of 0.001 sec. was chosen for the calculations.

4. RESULTS AND DISCUSSION

Figure 1(a) shows the normal velocity history at the centerline of the wake for $Re = 14000$ at five step heights downstream. The power spectrum of the normal velocity fluctuations (Fig.1(b)) confirms the oscillatory nature of the flow with a single predominant frequency of about 4.7 Hz, which is in agreement with experimental results [8]. Figure 1(c) shows a marker particle trace at time=3 sec., which illustrates the shedding pattern. In order to calculate the time-mean kinetic energy of the velocity fluctuations, the fluctuating velocity component (organized + turbulent) is $\widehat{u_i} = u_i - \overline{U_i}$. For the 2-D plane geometry considered, the kinetic energy of the velocity fluctuations can be written as;

$$E = \frac{3}{4}\left(\widehat{u_1}^2 + \widehat{u_2}^2\right) \tag{12}$$

where $\widehat{u_i}^2 = u_i^2 - 2u_i\overline{U_i} + \overline{U_i}^2$, and the time-mean value of the kinetic energy of the velocity fluctuations is

$$\overline{E} = \frac{3}{4}\left(\overline{\widehat{u_1}^2} + \overline{\widehat{u_2}^2}\right) \tag{13}$$

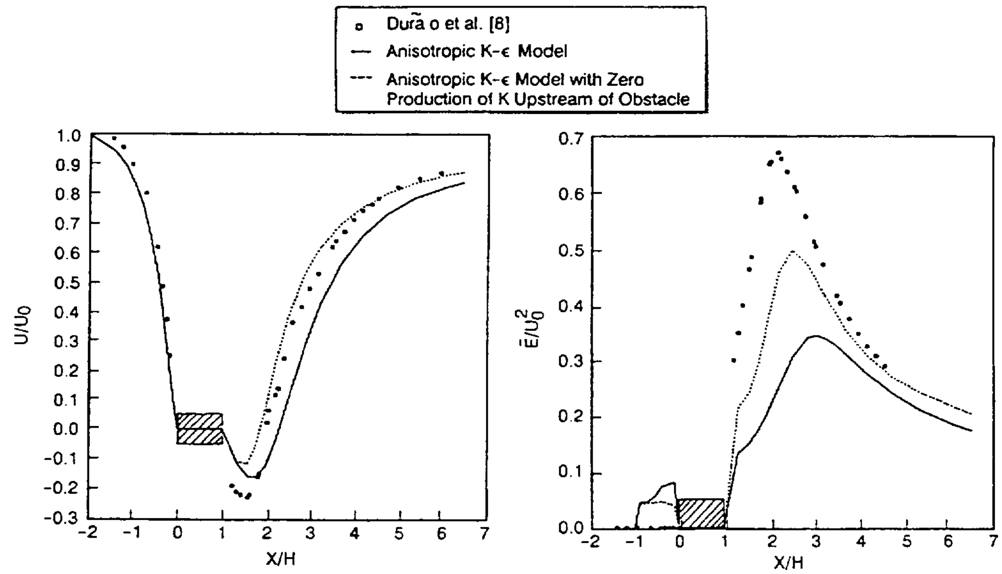


Fig.4: Centerline distribution of mean axial velocity

Fig.5: Time-mean kinetic energy of the velocity fluctuations

where

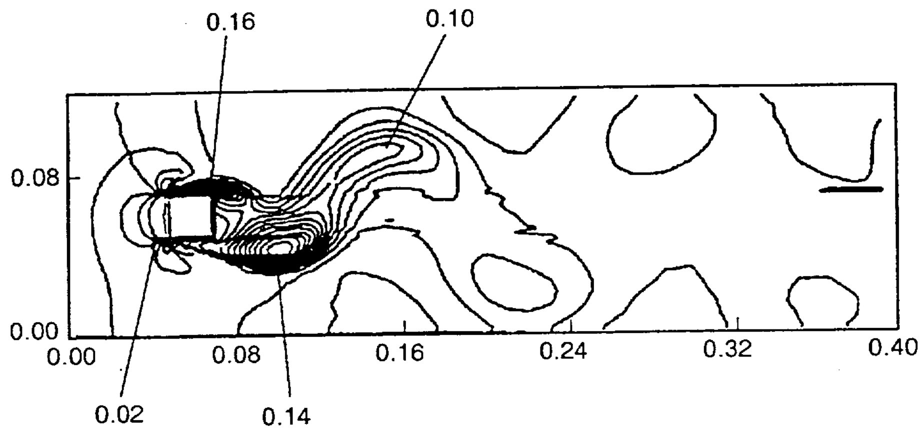
$$\overline{\widetilde{u}_i^2} = \left(\overline{u_i^2} - 2\overline{u_i\overline{U_i}} + \overline{U_i}^2\right) = \left[\overline{(U_i + u'_i)^2} - 2\overline{(U_i + u'_i)\,\overline{U_i}} + \overline{U_i}^2\right]$$

and from the definition of time averaging $\overline{u'_i U_i} = 0$, we get,

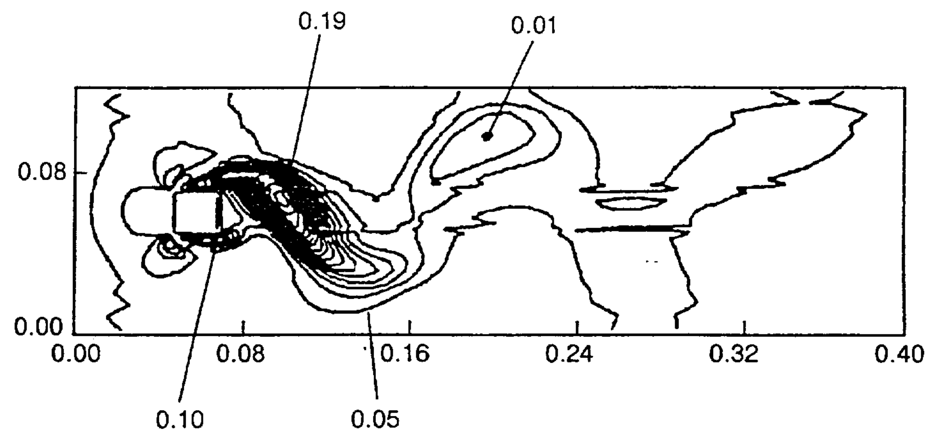
$$\overline{\widetilde{u}_i^2} = \overline{U_i^2} - \overline{U_i}^2 + \overline{u'^2_i} \qquad (i = 1,\ 2) \tag{14}$$

The first two terms on the right hand side of (14) represent the organized periodic energy contribution, while the last term represents the turbulent energy contribution.

Figure 2 shows the distribution of the mean axial velocity at the centerline. The anisotropic model gives better distribution downstream of the obstacle. Figure 3 compares the calculated distribution of the time-mean kinetic energy of the fluctuating motion (periodic + turbulent) along the centerline of the flow. The figure shows a better trend exhibited by the anisotropic k-ε model due to the improved resolution of the normal stresses. The standard k-ε model acts to damp the periodic fluctuations by producing too much eddy viscosity, which underestimates the time-averaged momentum transfer. Hence, the length of the separation region behind the obstacle is overpredicted. Also, the maximum of the kinetic energy at the centerline lies further downstream. The length of the recirculation zone and the location of the maximum fluctuating energy are improved by using the anisotropic model. The figure also shows some fluctuating energy in front of the obstacle, whereas measurements indicated that the flow remained virtually laminar there. This is because in the k-ε model the large velocity gradients at the stagnation region produce large turbulent kinetic energy. Results are also obtained from calculations in which the production of k in front of the obstacle was suppressed. Figure 4 shows the mean axial velocity distribution indicating better comparison with the experiment downstream of the obstacle. Figure 5 shows the distribution of the mean kinetic energy along the centerline. It can be seen that suppressing the production of the kinetic energy in front of the obstacle causes an



(a) Using the standard k-ε model



(b) Using the anisotropic k-ε model

Fig.6: $< v' v' > /U^2$ contours at $T = 3\,\mathrm{sec}$

increase in the fluctuating energy. Also, the peak of the energy fluctuations is shifted slightly upstream closer to the experimental data. The figure also shows smaller residual fluctuating energy in front of the obstacle. Figure 6(a) and (b) show the contour plots of the normal turbulent stress term $< v' v' >$ at an instant $T = 3\,\mathrm{sec}$. It can be seen that the anisotropic k-ε model produces higher $< v' v' >$ values, which act to increase the total fluctuating energy.

5. CONCLUSIONS

The turbulent vortex shedding flow behind a square cylinder was analyzed using an efficient time-accurate numerical method based on the PISO methodology. Turbulence was modeled using an anisotropic k-ε model which resolves the anisotropy of the Reynolds stresses reasonably well. Comparisons with the experimental data show the advantages of the model as compared with the standard isotropic k-ε model. Accurate predictions, however, can only be made by accounting for the history and transport effects of the individual Reynolds stresses. The anisotropic k-ε model seems to offer a compromise between the computationally intensive

Reynolds stress model and the standard isotropic k-ε model.

REFERENCES

[1] R.Davis and E.Moore (1982) A numerical study of vortex shedding from rectangles; J. Fluid Mech. vol.116 p.475

[2] M.Braza, P.Chassaing and H.Ha Minh (1986) Numerical study and physical analysis of the pressure and velocity fields in the near wake of a cylinder; J. Fluid Mech. vol.165 p.79

[3] R.Franke, W.Rodi and B.Schönung (1989) Analysis of experimental vortex shedding data with respect to turbulence modelling; 7th Symp. on Turbulent Shear Flows, Stanford Univ., Aug. 21, 1989

[4] B.Cantwell and D.Coles (1983) An experimental study of entrainment and transport in the turbulent near wake of a circular cylinder; J. Fluid Mech. vol.136 p.321

[5] J.MacInnes R.Claus and P.Huang (1989) Time-dependent calculation of a forced mixing layer using a k-ε turbulence model; 7th Symp. on Turbulent Shear Flows, Stanford Univ., Aug. 21, 1989

[6] I.Weisbrot and I. Wygnanski (1989) On coherent structures in a highly excited mixing layer; J. Fluid Mech., vol.195 p.321

[7] S.Majumdar and W.Rodi (1985) Numerical calculations of turbulent flow past circular cylinder; 3rd Symp. on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, California, 1985

[8] D.Durão, M.Heitor and J.Pereira (1988) Measurements of turbulent and periodic flows around a square cross-section cylinder; Experiments in Fluids, vol.6 p.298

[9] R.Issa (1986) Solution of the implicitly discretized fluid flow equations by operator-splitting; J. Comput. Phys., vol.62 p.40

[10] A.Yoshizawa (1984) Statistical analysis of the deviation of the Reynolds stress from its eddy-viscosity representation; Phys. Fluids, vol.27 p.1377

[11] C.Speziale (1987) On nonlinear k-l and k-ε models of turbulence; J. Fluid Mech., vol.178 p.459

[12] B.Launder and D.Spalding (1974) The numerical computation of turbulent flows; Comput. Meth. Appl. Mech. Engng., vol.3 p.269

[13] S.Nisizima and A.Yoshizawa (1987) Turbulent channel and Couette flows using an anisotropic k-ε model; AIAA J., vol.25 p.414

[14] H.Myong and N.Kasagi (1990) Prediction of anisotropy of the near-wall turbulence with an anisotropic low-Reynolds number k-ε turbulence model; J. Fluid Engng., vol.112 p.521

[15] W.Jones and B.Launder (1973) The calculation of low-Reynolds number phenomena with a two-equation model of turbulence; Int. J. Heat Mass Transfer, vol.16 p.1119

[16] H.Chen and V.Patel (1988) Near-wall turbulence models for complex flows including separation; AIAA J., vol.26 p.641

[17] A.Hadid, D.Chan, R.Issa and M.Sindir (1988) Convergence and accuracy of pressure-based finite difference schemes for incompressible viscous flow calculations in nonorthogonal coordinate system; AIAA Paper 88-3529

Single point modeling of rotating turbulent flows

By A. H. Hadid[1], N. N. Mansour[2] AND O. Zeman[3]

A model for the effects of rotation on turbulence is proposed and tested. These effects which influence mainly the rate of turbulence decay are modeled in a modified turbulent energy dissipation rate equation that has explicit dependence on the mean rotation rate. An appropriate definition of the rotation rate derived from critical point theory and based on the invariants of the deformation tensor is proposed. The modeled dissipation rate equation is numerically well behaved and can be used in conjunction with any level of turbulence closure. The model is applied to the two-equation k-ϵ turbulence model and is used to compute separated flows in a backward-facing step and an axisymmetric swirling coaxial jets into a sudden expansion. In general, the rotation modified dissipation rate model show some improvements over the standard k-ϵ model.

1. Motivation and objectives

The ability to accurately model the effects of rotation on turbulence has a wide variety of important applications in rotating machinery and combustion devices. Many turbulent flows of engineering importance involve combinations of rotational and irrotational strains. However, turbulence models of the eddy viscosity type are oblivious to the presence of rotational strains since they depend only on the mean velocity gradients through their symmetric part (i.e. the mean rate of strain tensor). The rotation rate, for example, does not explicitly enter the equations for the turbulent kinetic energy and its dissipation rate, yet evidence from experiments (Wigeland and Nagib 1978, Jacquin et al. 1990) and from direct numerical simulation (Bardina *et al.* 1985, Speziale *et al.* 1987, Mansour *et al.* 1991) show that the decay rate of turbulence is reduced by the presence of rotation.

The effects of rotation on turbulence are known to be subtle. They are manifested through changes in the spectrum of the turbulence caused by nonlinear interactions. For initially isotropic turbulence, rotation inhibits the cascade of energy from large to small scales. Zeman (1994) proposed a modified energy spectrum that takes into account the effects of rotation at high Reynolds number by introducing a rotation wavenumber, $k_\Omega = \sqrt{\Omega^3/\epsilon}$, below which rotation effects on spectral transfer are important. Much of the application work in simulating rotating flows have been conducted using varieties of eddy viscosity models (k-ϵ or k-l) and second order closure models with modified dissipation rate transport equation to account for

1 Rocketdyne Division/Rockwell International

2 NASA/Ames Research Center

2 Center for Turbulence Research

rotational effects. However, most of these models fail to predict the asymptotic behavior of the turbulence decay rate in the limits of large rotation rate. The objectives of this work are to model the effects of rotation using single-point two equation models and to offer an appropriate definition of the mean rotation rate that is consistent with the fact that spin is the main cause of reduction in the dissipation rate.

2. Accomplishments

For incompressible viscous flow with constant properties, the modeled transport equations for the turbulent kinetic energy, k, and its dissipation rate, ϵ, that are widely used for engineering applications take the form;

$$k_{,t} + U_j k_{,j} = D_k + P_k - \epsilon \tag{1}$$

$$\epsilon_{,t} + U_j \epsilon_{,j} = D_\epsilon + P_\epsilon - \Phi_\epsilon \tag{2}$$

where D_k and D_ϵ are the diffusion terms for k and ϵ respectively and are modeled as

$$D_k = \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right) k_{,j}\right]_{,j}, \qquad D_\epsilon = \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right) \epsilon_{,j}\right]_{,j},$$

where ν is the laminar viscosity and ν_t is the eddy viscosity $= C_\mu k^2/\epsilon$. σ_k and σ_ϵ are the ratio of Prandtl to Schmidt numbers and are taken as constants. P_k is the production term for k given as $P_k = -\overline{u_i' u_j'} U_{i,j}$, where $\overline{u_i' u_j'}$ is the Reynolds stress term and U_i is the mean velocity in the i-direction.

Assuming that the production of the dissipation rate P_ϵ is proportional to the production of turbulent kinetic energy P_k, i.e $P_\epsilon \sim P_k/T$ where T is the turbulent time scale given by $T = k/\epsilon$. Similarly assume that the destruction rate of dissipation rate Φ_ϵ is proportional to the turbulent energy dissipation rate term, i.e. $\Phi_\epsilon \sim \epsilon/T$. The modeled form of the dissipation rate equation becomes

$$\epsilon_{,t} + U_j \epsilon_{,j} = D_\epsilon + C_1 \frac{\epsilon}{k} P_k - C_2 \frac{\epsilon^2}{k} \tag{3}$$

Due to the symmetry of the Reynolds stress tensor $\overline{u_i' u_j'}$, the kinetic energy production term can be written as $P_k = -\overline{u_i' u_j'} S_{ij}$, where $S_{ij} = (U_{i,j} + U_{j,i})/2$ is the mean rate of strain tensor. Therefor it can be seen that the standard dissipation rate, eq. (3), has no explicit dependence on the mean rotation tensor $\Omega_{ij} = (U_{i,j} - U_{j,i})/2$. It follows that the commonly used modeled dissipation rate equation can only be affected indirectly by rotational strains through the changes that they induce in the Reynolds stress tensor.

In order to sensitize the dissipation rate equation to rotational effects, consider the simple case of isotropic turbulence in a rotating frame. In this case, an initially decaying isotropic turbulence is described by;

$$k_{,t} = -\epsilon \tag{4}$$

$$\epsilon_{,t} = -C_2 \frac{\epsilon^2}{k} \tag{5}$$

Equations (4) and (5) do not distiguish the difference between isotropic turbulence in a rotating frame and in an inertial frame. Models that have a non zero rotational correction have been proposed by Bardina *et al.* (1985), for example, for rotating isotropic turbulence where eq. (5) takes the form

$$\epsilon_{,t} = -C_2 \frac{\epsilon^2}{k} - C_3 \Omega \epsilon \tag{6}$$

with $C_2 = 1.83$ and $C_3 = 0.15$.

The above model is able only to accurately predict the reduction in the decay rate of the turbulent kinetic energy in rotating isotropic turbulence for weak to moderate rotation rates where the effects are small. However, for sufficiently high rotation rates and long enough time, the model drastically underpredicts the decay rate of the turbulent kinetic energy.

Hanjallic and Launder (1980) proposed a model for which the ϵ-transport equation in rotating isotropic turbulence takes the form

$$\epsilon_{,t} = -C_2 \frac{\epsilon^2}{k} - C_3 \Omega^2 k \tag{7}$$

where $C_2 = 1.92$ and $C_3 = 0.27$.

This model predicts unphysical behavior of negative dissipation rate at high rotation rates, thus violating the realizability constraint. Other modifications to the dissipation rate transport equation have been proposed to account for rotational strains, e.g Raj (1975) and Pope (1978). Again they fail in one way or another to account accurately for the rotational effects.

3. Proposed model

In the present work a new model is proposed that accounts for rotational effects and correctly predicts the asymptotic behavior at zero to inifinte rotation rates. Consider the dissipation rate equation in rotating isotropic turbulence

$$\epsilon_{,t} = -\left(1.7 + \frac{5}{6} \frac{\alpha^2}{\alpha^2 + 1}\right) \frac{\epsilon^2}{k} \tag{8}$$

with

$$\alpha = 0.35 Ro^{-1} \tag{9}$$

where Ro is the Rossby number defined as $Ro^{-1} = \Omega k/\epsilon$. For $\Omega >> 1$, $C_2 = 2.5$, which gives a power law exponent $n = 0.6$ (in $k \sim t^{-n}$) maching the power law proposed by Squires *et al.* (1993) for the asymptotic state of rotating homogeneous turbulence at high Reynolds numbers.

The experminental data of Jacquin *et al.* (1990) are used to test the proposed model. Their experiments consisted of measuring the velocity field and characteristic quantities characterizing the fluctuating field downstream of a rotating cylinder containing a honycomb structure and a turbulence producing grid. The coupled differential equations for k and ϵ describing the effects of rotation on an initially isotropic turbulence can be written as

$$k_{,t} = -\epsilon \tag{10}$$

$$\epsilon_{,t} = -\left(C_2 + C_3\frac{\alpha^2}{\alpha^2+1}\right)\frac{\epsilon^2}{k} \tag{11}$$

These equations were solved numerically using a fourth-order Runge-Kutta integration scheme. The model predictions (with $C_2 = 1.7$ and $C_3 = 5/6$) are compared with the experimental data of Jacquin *et al.* (1990) as shown in Fig. 1a. The model predicts well the evolution of turbulent kinetic energy and its decay rate for a wide range of rotation rates. We have also tested the model for the three Reynolds numbers measured by Jacquin *et al.* (1990), and found similar agreement of the model predictions with the data. We should point out at this point that the value $C_2 = 1.7$, proposed here for zero rotation rate, is lower than the value conventionally used in k-ϵ modeling. We find that with the conventional value of $C_2 = 1.92$ (and $C_3 = 3/5$) the model fails to predict the experimental data (see Fig. 1b)

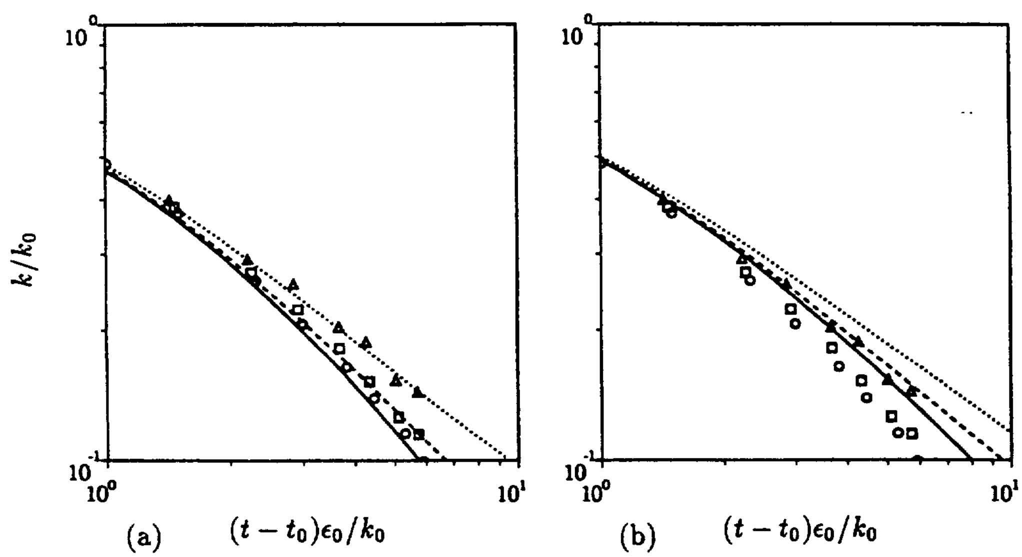


FIGURE 1. Decay of turbulent kinetic energy. Symbols are the data of Jacquin *et al.* (1990), lines are the model predictions. ○ & ——— $\Omega = 62.8$ (rad/s), □ & ---- $\Omega = 31.4$ (rad/s), △ & ········ $\Omega = 15.7$. (a) Model predictions with $C_2 = 1.7$ and $C_3 = 5/6$; (b) Model predictions with $C_2 = 1.92$ and $C_3 = 3/5$.

4. Rotation Rate For General Flows

In order to test the rotational correction proposed in eq. (8) to the dissipation rate equation for general flows where the rotation rate is a function of position and in the presence of mean strains, the question arises as to what is the appropriate definition of the rotation rate, Ω?

In most previous studies, the rotation rate or the mean vorticity Ω was replaced by $\sqrt{\Omega_{ij}\Omega_{ij}/2}$, where $\Omega_{ij} = (U_{i,j} - U_{j,i})/2$ is the rotation rate tensor of the mean flow. However, such definition does not distinguish between a vortex sheet and a vortex. A definition of a vortex or a region of vorticity (with spin) was given by Chong *et al.* (1990) -using the arguments of the critical point theory and the invariants of the deformation tensor- as a region in space where the vorticity is sufficiently strong to cause the rate of strain tensor to be dominated by the rotation tensor, i.e. the rate of deformation tensor has complex eigenvalues. This definition satisfies the principle of frame invariance since it depends only on the properties of the deformation tensor. We shall use it because the reduction in the dissipation rate is due mainly to the spin that the mean imposes on the turbulence. Consider the matrix D_{ij} of the elements of the deformation tensor,

$$D_{ij} = U_{i,j} \tag{12}$$

which can be split to

$$D_{ij} = S_{ij} + \Omega_{ij} \tag{13}$$

The complex eigenvalues of D_{ij} are found by solving the characteristic equation $|D_{ij} - \lambda\delta_{ij}| = 0$, where the λ's are the eigenvalues of D_{ij}. For a 3×3 matrix, λ can be found from the solution of

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0 \tag{14}$$

where P, Q and R are the matrix invariants and are given by

$$P = -U_{i,i} \tag{15}$$

$$Q = \frac{1}{2}(P^2 - S_{ij}S_{ji} - \Omega_{ij}\Omega_{ji}) \tag{16}$$

$$R = \frac{1}{3}\left(-P^3 + 3PQ - S_{ij}S_{jk}S_{ki} - 3\Omega_{ij}\Omega_{jk}S_{ki}\right) \tag{17}$$

For an incompressible flow $P = 0$ from continuity and the characteristic equation becomes

$$\lambda^3 + Q\lambda + R = 0 \tag{18}$$

Now if

$$A = \left[-\frac{R}{2} + \sqrt{\left(\frac{R^2}{4} + \frac{Q^3}{27}\right)}\right]^{1/3}$$

and,

$$B = \left[-\frac{R}{2} - \sqrt{\left(\frac{R^2}{4} + \frac{Q^3}{27}\right)}\right]^{1/3}$$

then the three roots of λ are;

$$\left[A+B, -\frac{A+B}{2}+\mathrm{i}\frac{A-B}{2}\sqrt{3}, -\frac{A+B}{2}-\mathrm{i}\frac{A-B}{2}\sqrt{3}\right]$$

That is λ can have:

(i) all real roots which are distinct when

$$\left[(Q/3)^3+(R/2)^2\right]<0,$$

or

(ii) all real roots where at least two roots are equal when

$$\left[(Q/3)^3+(R/2)^2\right]=0,$$

or

(iii) one real root and a pair of complex conjugate roots when

$$\left[(Q/3)^3+(R/2)^2\right]>0.$$

We shall follow Chong *et al.* (1990) and define the rotation rate

$$\Omega=\Im(\lambda)=\frac{\sqrt{3}}{2}(A-B), \quad \text{when } \left[(Q/3)^3+(R/2)^2\right]>0, \tag{19}$$

$\Omega=0$ otherwise. It is important to note that for two dimensional Cartesian flows, the rotation rate defined by Eq. (19) reduces to $\Omega=\sqrt{|Q|}$, when Q, the determinant of the deformation tensor matrix, is negative. For pure shear the definition, eq. (19) yields $\Omega=0$. Conventional models that are calibrated for shear flows, need not be recalibrated when corrections based on Ω are added to the model.

5. Numerical Procedure

For a two-dimensional, incompressible and steady turbulent flow, the Reynolds averaged momentum, continuity, turbulent kinetic energy and dissipation rate equations can be written in the generalized form;

$$\frac{\partial}{\partial x}(\rho U\Phi)+\frac{1}{r}\frac{\partial}{\partial y}(\rho rV\Phi)=\frac{\partial}{\partial x}\left(\Gamma_{\Phi,x}\frac{\partial\Phi}{\partial x}\right)+\frac{1}{r}\frac{\partial}{\partial y}\left(r\Gamma_{\Phi,r}\frac{\partial\Phi}{\partial y}\right)+S_\Phi \tag{20}$$

Where $r=1$ for Cartesian two-dimensional flow, and $y=r$ for two-dimensional axisymmetric flow. Table 1 gives a summary of the terms in eq. (20) for the dependent variables solved in the code.

Φ	Γ_{Φ_x}	Γ_{Φ_r}	S_Φ
1	0.	0.	0.
U	$2\mu_e$	μ_e	$-\partial P/\partial x + 1/r\partial(\mu_e r\partial V/\partial x)/\partial y$
V	μ_e	$2\mu_e$	$-\partial P/\partial y + \partial(\mu_e\partial U/\partial y)/\partial y$
W	μ_e	μ_e	$-\rho VW/r - W/r^2\partial(r\mu_e)/\partial r$
k	$\mu + \mu_t/\sigma_k$	$\mu + \mu_t/\sigma_k$	$P_k - \rho\epsilon$
ϵ	$\mu + \mu_t/\sigma_\epsilon$	$\mu + \mu_t/\sigma_k$	$C_1 P_k\epsilon/k - C_2\rho\epsilon^2/k$

Table 1. Summary of the governing equations. ρ is the density, Γ_{Φ_x} and Γ_{Φ_r} are the exchange coefficients in the axial and radial directions respectively, S_Φ is the source term for the variable Φ. In the table, μ_e is the effective viscosity given as $\mu_e = \mu + \mu_t$, where μ is the laminar viscosity and μ_t is the turbulent viscosity, $\mu_t = C_\mu\rho k^2/\epsilon$.

In the standard k-ϵ turbulence model the constants C_μ, C_1, C_2, σ_k and σ_ϵ have the values 0.09, 1.44, 1.92, 1.0 and 1.0 respectively.

In the rotation modified k-ϵ turbulence model, only C_2 takes the form given by eq. (11) i.e, $C_2 = 1.7 + (5/6)\alpha^2/(\alpha^2+1)$.

The governing transport eq. (20) is solved using the primitive variables on a nonstaggered mesh and converted into a system of algebraic equations by integrating over control volumes defined around each grid point. The SIMPLE pressure-correction scheme (Patankar 1980) is used to couple the pressure and velocities and the resulting algebraic equations are solved iteratively. The convective terms are differenced using a second-order upwind scheme while the diffusion terms are approximated by a central differencing scheme.The physical domain is discretized using a non-uniform mesh where grid points are clustered close to the walls.

6. Model Application

The performance of the present model for complicated recirculating flows is demonstrated through calculations and comparisons with the experimental data of Driver & Seegmiller (1985) for backward-facing step flows and with the experiments of Roback & Johnson (1983) for a confined swirling coaxial jets into a sudden expansion.

Figure 2, shows the streamlines for the backward-facing step using the rotation modified k-ϵ turbulence model. The calculations were performed on a 100x40 grid points. The computational domain had a length of $50H$ (H is the step height) and a width of $9H$. The experimental data were used to specify the inflow conditions for a channel flow calculation where the fully developed profiles at the channel exit were used as the inlet conditions for the backward-facing step calculations. Fully developed flow conditions were imposed at the outflow boundary. The standard wall function approach (Launder & Spalding 1974) was used to bridge the viscous sublayer near the wall.

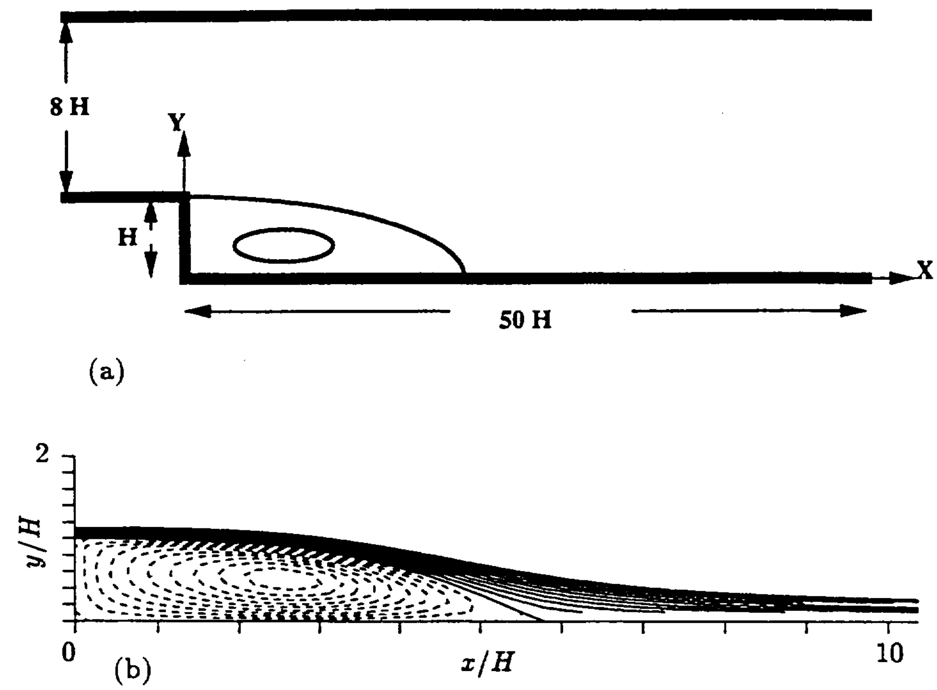


FIGURE 2. Backward-facing step geometry and stream-function contours. The contour levels were set between (-0.1 and 0.1) with an increment level = 0.01. ---- negative values, —— positive values.

The computed reattachment lengths were $5.50H$ using the standard k-ϵ turbulence model and $6.22H$ for the rotation modified k-ϵ turbulence model. The modified k-ϵ model prediction is closer to the experimental value of $6.10H$. While these results are encouraging, they are mainly due to the fact that we have changed the value of C_2 for the non-rotating case. In general, a change in the value of C_2 will result in poor predictions of the mean profiles. The mean velocity profile at three locations downstream are shown on Fig. 3, while the turbulent stress profiles at $X/H = 4$ are shown on Fig. 4. All the quantities were normalized by the step height (H) and the experimental reference free-stream velocity (U_{ref}). It can be seen that the overall performance of the rotation modified dissipation rate equation is better than the standard k-ϵ model especially in the recirculation region (Figs. 3a, and 4). Some improvements are also obtained in the recovery region using the modified k-ϵ model. Figure 5 shows the contours of the effective rotation rate used as defined by Eq. (19).

For the 2D/axisymmetric swirling flow computations, the expressions for the invariants Q and R (Eqs. (16) & (17) respectively) are expanded and Eq. (19) is used to obtain the values of Ω. The model was used to predict the mean profiles for a confined double concentric jets with a swirling outer jet flow into a sudden expansion (Roback & Johnson, 1983, see Fig. 6). Measurements are available for the mean

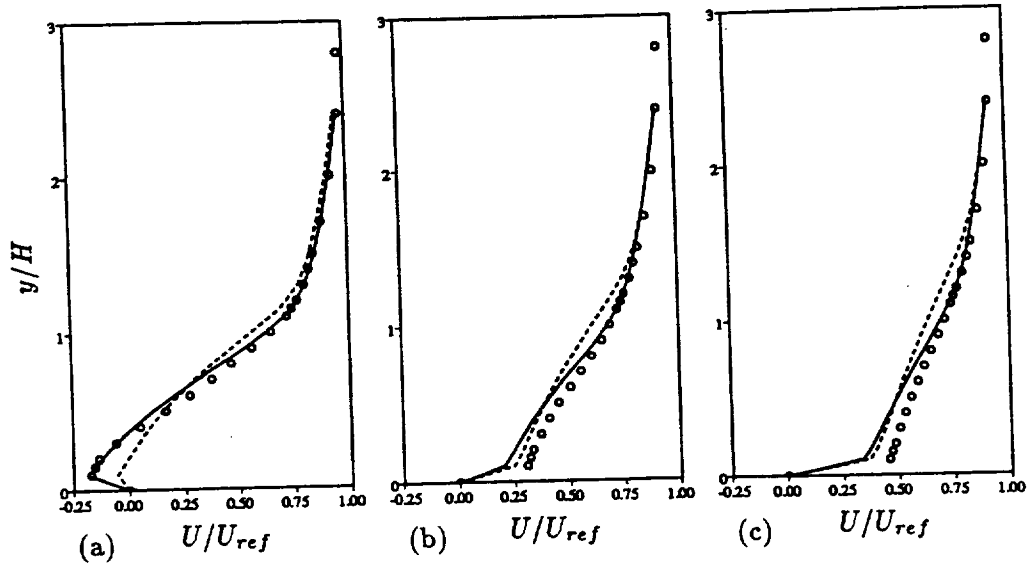


FIGURE 3. Mean axial velocity profiles at different axial locations. o data (Driver & Seegmiller, 1985); —— modified k-ϵ model; ---- standard k-ϵ model. (a) $X/H = 4$, (b) $X/H = 8$, (c) $X/H = 12$.

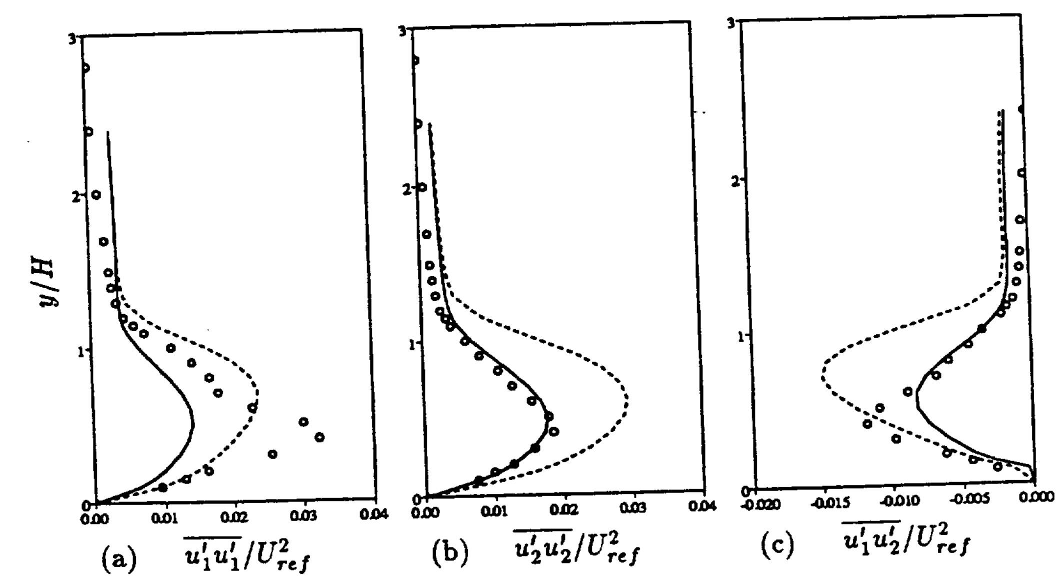


FIGURE 4. Turbulent stress profiles at $X/H = 4$. o data (Driver & Seegmiller, 1985); —— modified k-ϵ model; ---- standard k-ϵ model. (a) $\overline{u_1'u_1'}/U_{ref}^2$, (b) $\overline{u_2'u_2'}/U_{ref}^2$, (c) $\overline{u_1'u_2'}/U_{ref}^2$.

velocity profiles and velocity fluctuations downstream of the expansion. Simulations with a coarse nonuniform grid of 30×20 mesh points were made. However, there is some uncertainty about the inlet conditions to be used since the first velocity

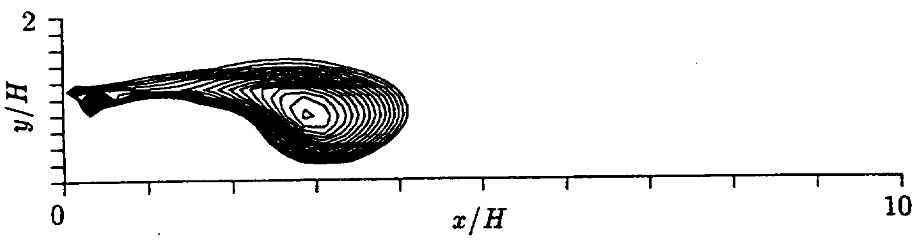


FIGURE 5. Contours of the effective rotation rate, Ω. Contour levels were set between (0.1,1.0) with an increment level = .01. are

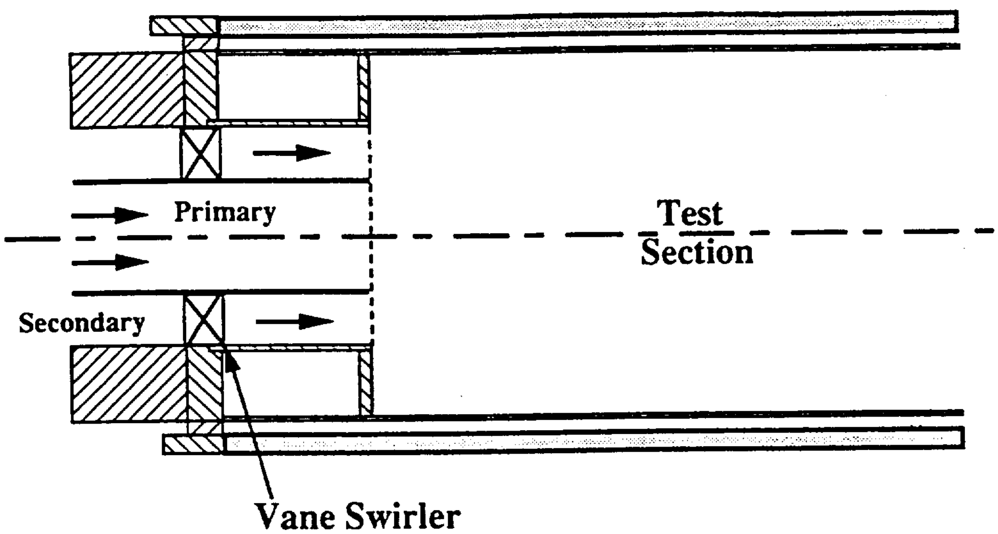


FIGURE 6. Roback & Johnson's swirling coaxial jets discharging into an expanded duct.

profiles measured were 5mm downstream of the expansion.

To predict this flow, the measured profiles at 5mm were adjusted near the edges and were used as inlet conditions at the expansion plane. Preliminary results obtained with the coarse mesh indicate similar trends as the experiment. Figure 7 shows the streamline contours using the standard and the modified k-ϵ turbulence models. The figure shows that a closed internal recirculation zone forms at the center with an additional zone at the corners downstream of the step. This causes a flow diversion outwards with high gradients between these regions. Figure 8 shows the axial and tangential velocity profiles at 25 mm downstream of the expansion using the standard and the modified k-ϵ turbulence models. Results in this case indicate little or no improvements offered using the modified k-ϵ model over the standard k-ϵ model. Finer mesh may improve the results but the uncertainties in the inlet boundary conditions raise the question about the adequacy of using this experiment for validation purposes.

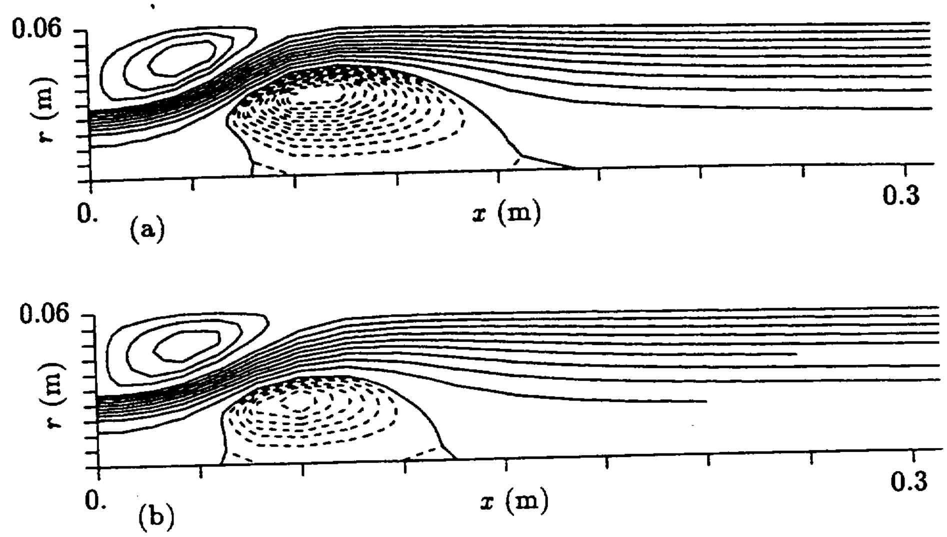


FIGURE 7. Swirling coaxial jets discharging into an expanded duct. Stream-function contour. ---- levels were set between (-0.15,0.) with an increment level = 0.01, —— levels were set between (0.,0.7) with an increment level = 0.05. (a) Standard k-ϵ model, (b) Modified k-ϵ model.

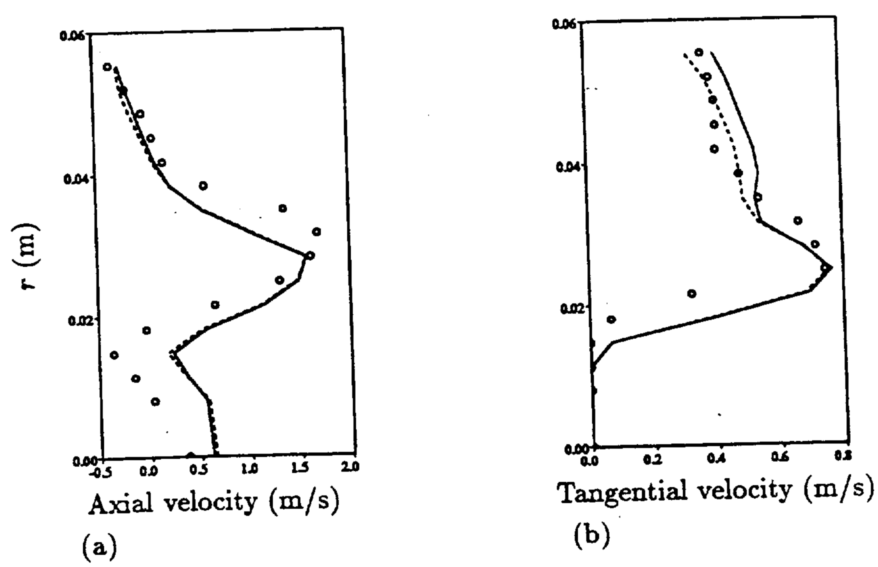


FIGURE 8. Velocity profiles at $X = 25$ mm. o data (Roback & Johnson, 1983); —— modified k-ϵ model; ---- standard k-ϵ model. (a) Axial Velocity, (b) Tangential velocity.

7. Conclusions

A new simple model for the turbulent energy dissipation rate equation has been proposed to account for the rotational effects on turbulence. A frame invariant definition of the rotation rate proposed by Chong *et al.* (1990) based on the critical point theory was used. The model can be used in conjunction with any level of turbulence closure. It was applied to the two-equation k-ϵ turbulence model and was tested for separted flows in a backward-facing step and for axisymmetric swirling jet into a sudden expansion. The model is numerically stable and showed improvements over the standard k-ϵ turbulence model. It is important to point out that the present study was carried out to roughly evaluate the model, but that a systematic recalibration of the constants in the k-ϵ model is needed before going any further with the proposed model.

The authors would like to acknowledge many discussions with Dr. K. Shariff regarding proper definition of the rotation rate.

REFERENCES

BARDINA, J., FERZIGER, J. & ROGALLO, R. 1985 Effects of Rotation on Isotropic Turbulence: Computation and Modeling. *J. Fluid Mech.* **154,** 321-336.

DRIVER, D. & SEEGMILLER, H. 1985 Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow. *AIAA Journal.* **23,** 163-171.

HANJALIC, K. & LAUNDER, B. 1980 Sensitizing the Dissipation Equation to Irrotational Strains. *ASME J. Fluids Eng.* **102,** 34-40.

JACQUIN, L., LEUCHTER, O., CAMBON, C. & MATHIEU J. 1990 Homogenous Turbulence in the Presence of Rotation. *J. Fluid Mech.* **220,** 1-52.

LAUNDER, B. & SPALDING, D. 1974 The Numerical Computation of Turbulent Flows. *Comput. Methods Appl. Mech. and Engg.* **3,** 269-289.

MANSOUR, N. N., CAMBON, C. & SPEZIALE C. G. 1991 Theoretical and computational study of rotating isotropic turbulence. *Studies in Turbulence,* ed. by T. B. Gatski, S. Sarkar and C. G. Speziale, Springer Verlag, New-York.

CHONG, M. S., PERRY, A. E. & CANTWELL, B. J. 1990 A General Classification of Three-dimensional Flow Fields. *Phys. Fluids A.* **2,** (5), 765-777.

ROBACK, R. & JOHNSON, B. 1983 Mass and Momentum Turbulent Transport Experiment With Confined Swirling Co-Axial Jets. *NASA CR-168252.*

SPEZIALE, C., MANSOUR, N. & ROGALLO, R. 1987 The Decay of Isotropic Turbulence in a Rapidlly Rotating Frame. *Proceedings of the 1987 Summer Program,*CTR, NASA Ames/Stanford University.

SQUIRES, K., CHASNOV, J., MANSOUR, N. & CAMBON, C. 1993 Investigation of the Asymptotic State of Rotating Turbulence Using LES. *Annual Research Briefs - 1990,* CTR, NASA Ames/Stanford University.

WIGELAND, R. & NAGIB, H. 1978 Grid-Generated Turbulence With and Without Rotation About the Streamwise Direction. *IIT Fluids and Heat Transfer Report, R78-1,* Illinois Institute of Technology.

ZEMAN, O. 1994 A Note on the Spectra and Decay of Rotating Homogeneous Turbulence. *Phys. Fluids.* **6,** 3221-3223.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
Title and Subtitle **Comapartive Study of Advanced Turbulence Models for Turbomachinery**		5. Report Date **October 1996**
7. Author(s) **Ali H. Hadid and Munir M. Sindir**		6. Performing Organization Code
9. Performing Organization Name and Address **Rocketdyne Division/Rockwell International** **6633 Canoga Avenue, P. O. Box 7922** **Canoga Park, CA 91303-7922**		8. Performing Organization Report No. 10. Work Unit No.
12. Sponsoring Agency Name and Address **National Aeronautics and Space Adminstration** **George C. Marshall Space Flight Center** **Huntsville, AL 35812**		11. Contract or Grant No. **NAS8-38860** 13. Type of Report and Period Covered **Final** **Feb. 1992 - Oct. 1995**
15. Supplementary Notes **Project Monitors, Dr. Paul McConnaughey and Mr. Robert Garcia, NASA/MSFC** **Program Manager, Dr. Glenn Havskjold, Rocketdyne Division/Rockwell International**		14. sponsoring Agency Code

16. Abstract

A computational study has been undertaken to study the performance of advanced phenomenological turbulence models coded in a modular form to describe incompressible turbulent flow behavior in two dimensional/axisymmetric and three dimensional complex geometry. The models include a variety of two equation models (single and multi-scale k -ε models with different near wall treatments) and second moment algebraic and full Reynolds stress closure models. These models were systematically assessed to evaluate their performance in complex flows with rotation, curvature and separation. The models are coded as self contained modules that can be interfaced with a number of flow solvers. These modules are stand alone satellite programs that come with their own formulation, finite-volume discretization scheme, solver and boundary condition implementation. They will take as input (from any generic Navier-Stokes solver) the velocity field, grid (structured H-type grid) and computational domain specification (boundary conditions), and will deliver, depending on the model used, turbulent viscosity, or the components of the Reynolds stress tensor. There are separate 2D/axisymmetric and/or 3D decks for each module considered.

The modules are tested using Rocketdyn's proprietary code REACT. The code utilizes an efficient solution procedure to solve Navier-Stokes equations in a non-orthogonal body-fitted coordinate system. The differential equations are discretized over a finite-volume grid using a non-staggered variable arrangement and an efficient solution procedure based on the SIMPLE algorithm for the velocity-pressure coupling is used. The modules developed have been interfaced and tested using finite-volume, pressure-correction CFD solvers which are widely used in the CFD community. Other solvers can also be used to test these modules since they are independently structured with their own discretization scheme and solver methodology. Many of these modules have been independently tested by Professor C.P. Chen and his group at the University of Alabama at Huntsville (UAH) by interfacing them with own flow solver (MAST).

17. Key Words (Suggested by Author(s)) **Turbulence Module decks, Swirling Flow, Turbulent Transport**	18. Distribution Statement **Unclassified - Unlimited**

19. Security Classif. (of this report) **Unclassified**	20. Security Classif. (of this page) **Unclassified**	21. No. of pages	22. Price*

NASA FORM 1626 OCT 88

*For sale by the National Technical Information Service, Springfield, Virginia 22161